Combinatorial and Compositional Aspects of Bilingual Aligned Corpora

Spyros Martzoukos

Combinatorial and Compositional Aspects of Bilingual Aligned Corpora

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof. dr. ir. K.I.J. Maex
ten overstaan van een door het College voor Promoties ingestelde
commissie, in het openbaar te verdedigen in
de Aula der Universiteit
op vrijdag 21 oktober 2016, te 11:00 uur

door

Spyridon Martzoukos

geboren te Cholargos, Griekenland

Promotiecommissie

Promotor:		
	Prof. dr. M. de Rijke	Universiteit van Amsterdam
Co-promotor:		
	Dr. C. Monz	Universiteit van Amsterdam
Overige leden:		
	Prof. dr. J. Bos	Rijksuniversiteit Groningen
	Prof. dr. A.P.J. van den Bosch	Radboud Universiteit Nijmegen
	Dr. E. Kanoulas	Universiteit van Amsterdam
	Dr. M.J. Marx	Universiteit van Amsterdam
	Prof. dr. K. Sima'an	Universiteit van Amsterdam

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

SIKS Dissertation Series No. 2016-42
The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

The research was supported by CoSyne under project number FP7-ICT-4-248531.

Acknowledgments

Although the research for this thesis was carried out during 2010–2014, it is an outcome of an intellectual journey that lasted a decade and spanned three institutions, namely the School of Mathematical Sciences at Queen Mary, University of London, the Department of Mathematical Science at the Univerity of Bath, and the Informatics Institute at the Univeristy of Amsterdam. I would thus like to thank all the people with whom I worked closely throughout these years (in order of appearance): Reza Tavakol, Shaun Bullett, Christian Beck, Peter Dobcsányi, Chris Budd, Jonathan Evans, Alastair Spence, Thomas Prellberg, Wolfram Just, Christof Monz and Maarten de Rijke.

Being part of the Information and Language Processing Systems (ILPS) group has been a great experience. Many thanks to the following ILPS members, each of which was special in his/her own way: Petra Best, Marc Bron, Christophe Costa Florêncio, Abdo El Ali, Katja Hofmann, Bouke Huurnink, Caroline van Impelen, Bogomil Kovachev, Edgar Meij, Christof Monz, Maarten de Rijke, Anne Schuth, Manos Tsagkias, Wouter Weerkamp and Masrour Zoghi. Special thanks to the powerful smt* and zookst* machines and to Jeroen Roodhart and Auke Folkerts for their superpowers.

I started writing this thesis in the summer of 2014 and it was completed by the autumn of 2015. Thank you Natalia for putting up with me during this period.

Contents

Chapter 1

Introduction

The integration of information via search engines and social media necessitates quick and accurate translation of online content. This task has revitalized the field of automated translation, which was in limbo up to the late 1990s. The (re)development of such translation mechanisms has been done exclusively to accomodate market requirements. Consequently, they are heuristic-driven and formal approaches towards an undesrtanding of multilinguality have been disregarded. In this thesis we take a step back and try to understand *why* things work.

Let $\mathcal{S}$ and $\mathcal{T}$ denote a source and target language vocabulary respectively. Machine Translation (MT) is the process of converting a string S consisting of words from $\mathcal{S}$ to a new string T consisting of words from $\mathcal{T}$. Each of S and T is assumed to be a sentence in a source and target natural language, respectively. As such their composition from vocabularies $\mathcal{S}$ and $\mathcal{T}$ is determined by additional structure, namely grammar and semantics [78]. The former dictates the ordering of words in a sentence, and such rules differ for each language [88]. On the other hand, semantics is a more abstract notion that confers meaning to sentences and is language independent [113]. String T is a translation of S if the following criteria are met:

- T obeys grammatical rules from the target language.
- The semantics of T is as close as possible to the semantics of S.

For a given source string S it is possible to have multiple target strings as appropriate translations. Depending on the extent to which the above criteria are met, it is possible to assess the quality of possible translation strings for any S. In Statistical Machine Translation (SMT) a candidate translation T for a given S is graded by a conditional probability $p(T|S)$. Translation

$$T^* = \arg\max_T \; p(T|S) \tag{1.1}$$

is the output of this MT process. Using Bayes' Theorem, (1.1) is rewritten as

$$T^* = \arg\max_T \; p(S|T)p(T), \tag{1.2}$$

which clearly reflects the semantic and grammaticality criteria mentioned above. This correspondence of $p(T)$ and $p(S|T)$ with the above criteria predisposes (1.2) to become the starting point of SMT. However, this is not done in practice and research in SMT focuses on directly modeling $p(T|S)$ in (1.1) [86].

At its core SMT fuses source and target grammatical correspondence with semantics in a probabilistic setting. This is achieved by automatically constructing a phrase-level source-to-target language dictionary with associated weights, or features, for each entry. Informally these features reflect

- the likelihood of existence of an entry in *any* sentence pair (S, T), and
- the likelihood of elementary local ordering structure of an entry in *any* sentence pair (S, T).

This dictionary is called a *phrase-table* [85]. It is memorized and used for translating any source sentence S into a target sentence T: this process, which is called *decoding*, breaks S into source phrases whose phrase-table entries are used to compose candidate target sentences $\{T\}$ with associated probabilities $\{p(T|S)\}$. The best translation is found according to (1.1).

This thesis is entirely devoted to providing further insights about the phrase-table and aspects related to it. In order to formulate our research directions we proceed with an introductory description of the steps that lead to the formation of the phrase-table.

1.1 From word alignments to the phrase-table

A parallel corpus, or simply a bitext, is a collection of source-target sentence pairs; each such pair (S, T) is such that S and T are translations of each other. A bitext is the input of most SMT systems and is used as training data from which translation rules are learned [150]. It is typically composed from multilingual resources that were manually translated such as parliamentary proceedings, patent documents, legal proceedings etc. Its size, i.e., the number of sentence pairs in the bitext, depends on the publicly available resources for the language pair and is typically in the range 500K–2.5M.

The first step of the training process is the automated extraction of building blocks of translation rules from the bitext. This is achieved by identifying the most likely correspondences between fragments in each sentence pair. This task is known as sequence alignment [62] and the so-called IBM models [19] provide the necessary tools for doing that: The bitext is treated as a bag of sentence pairs $\{(S_i, T_i)\}_{i=1}^N$ whose likelihood $\prod_{i=1}^N p(T_i|S_i)$ is maximized. Each (S_i, T_i) is assumed to be a bilingual word sequence. Information about source-to-target word alignments is encoded as a latent variable and is the output of this program. More specifically, since this variable is treated as a random function, each word in T_i is eventually aligned with at most one word from S_i. Similarly, in order to identify target-to-source word alignments, the likelihood $\prod_{i=1}^N p(S_i|T_i)$ is maximized. These two outputs are then combined into a

single word alignment using various heuristics. Throughout this thesis, the so-called 'grow-diag-final-and' method [84] is used. An example of typical such word-aligned sentence pairs are shown in Figure 1.1. The types of resulting word-alignments can be summarized as follows:
i) An alignment connects a source word to a target word and each such word is not connected to any another word via another alignment. Each of these words may or may not be in base linguistic form. This type of word alignment gives rise to a generalized dictionary entry, which is the most basic building block of translation rules [109].
ii) If a word w from either the source or target side has more than one alignment then each alignment of w may not necessarily reflect meaningful correspondences when observed in isolation. In this case, the extracted translation rule that includes w is of phrase-level, i.e., composed of multiple words in either the source or target side. More precisely, breadth-first search from w results in two sets, one consisting of source words and the other of target words. These sets together with the order of words in the sentence pair give rise to a phrase-level building block of translation rules.
iii) Certain words can be unaligned. Although a source and target sentence may be perfect translations of each other, it is possible to have a word, say, in the source side, as a necessary element for maintaining grammatical coherence, without any explicit correspondence in the target side. This is a consequence of what was mentioned in the beginning of this section: Two different grammars form two sentences that have to convey the same semantics.
iv) As with any Machine Learning method, training data may contain noise [24]. In the bitext noise is inaccurate or incomplete translations, which results in erroneous alignments in each sentence pairs. Even if the amount is small, alignment quality depends on the size of the bitext. A small bitext (∼500K sentence pairs) with at least one morphologically rich language results in alignments with excessive unaligned words.

The breadth-first search that was mentioned in ii) above is essentially the means of identifying all building blocks of translation rules. A type-i) building block is a special case of type-ii) and a type-iv) building block is simply an erroneous building block. For example, the building blocks of translation rules that are formed from the first sentence pair of Figure 1.1 are:

$$\begin{array}{rcl}
\texttt{im} & \leftrightarrow & \texttt{in} \\
\texttt{anschluss} & \leftrightarrow & \texttt{aftermath} \\
\texttt{an} & \leftrightarrow & \texttt{of} \\
\texttt{den 11. september} & \leftrightarrow & \texttt{9 / 11 ,} \\
\texttt{gab es} & \leftrightarrow & \texttt{there was} \\
\texttt{jede} & \leftrightarrow & \texttt{a} \\
\texttt{menge} & \leftrightarrow & \texttt{good deal of}
\end{array} \tag{1.3}$$

and similarly for the rest. An example of a phrase pair that fails to become a building

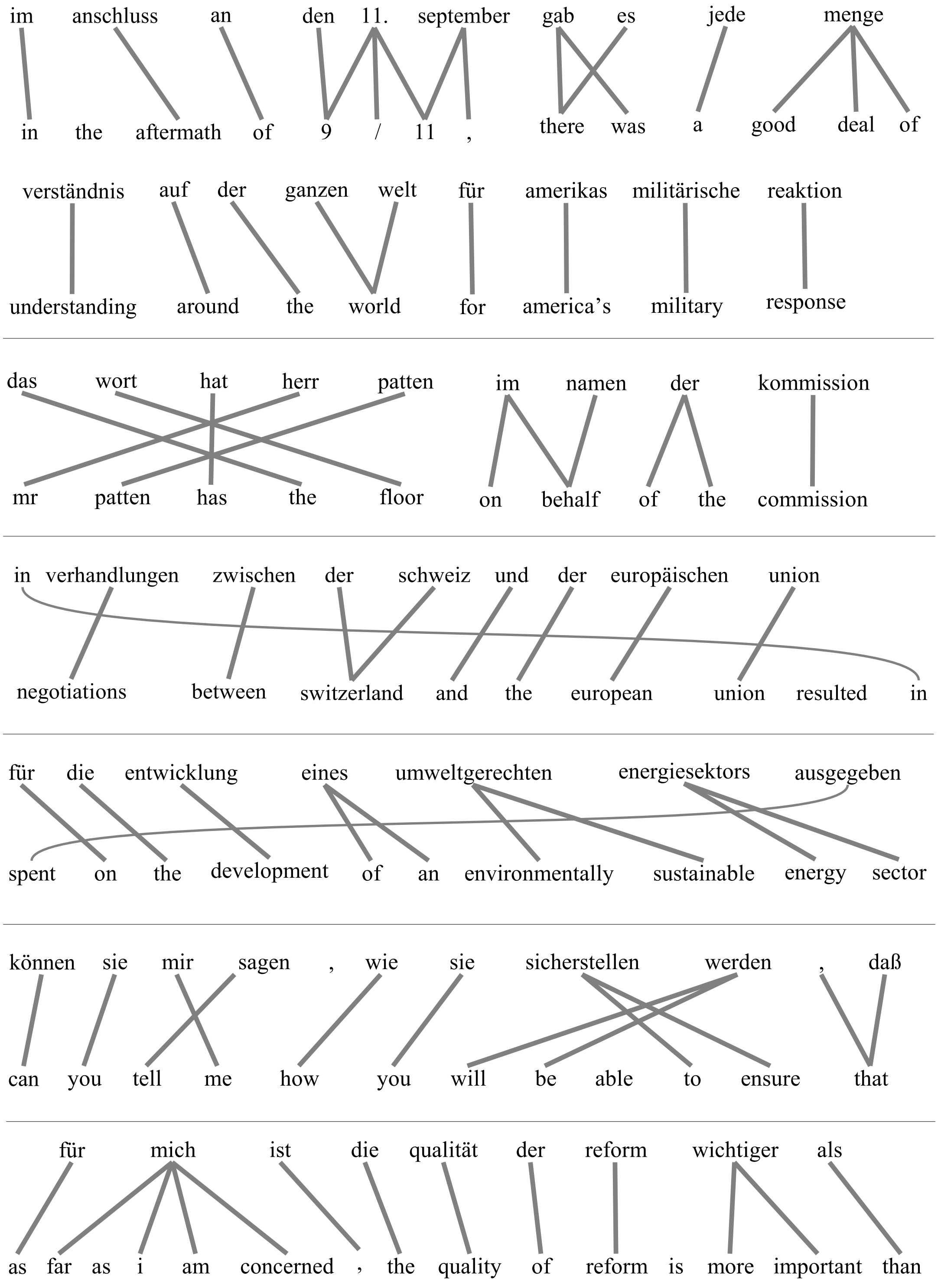

im anschluss an den 11. september gab es jede menge
in the aftermath of 9 / 11 , there was a good deal of
verständnis auf der ganzen welt für amerikas militärische reaktion
understanding around the world for america’s military response
das wort hat herr patten
mr patten has the floor
im namen der kommission
on behalf of the commission
in verhandlungen zwischen der schweiz und der europäischen union
negotiations between switzerland and the european union resulted in
für die entwicklung eines umweltgerechten energiesektors ausgegeben
spent on the development of an environmentally sustainable energy sector
können sie mir sagen , wie sie sicherstellen werden , daß
can you tell me how you will be able to ensure that
für mich ist die qualität der reform wichtiger als
as far as i am concerned , the quality of reform is more important than

Figure 1.1: Typical German-English word-aligned bitext.

block of translation rules is

$$\texttt{im anschluss an den} \leftrightarrow \texttt{in the aftermath of 9} \quad (1.4)$$

because it ignores certain words that are spanned from breadth-first search. In fact, this phrase pair even fails to become a (general) translation rule, i.e., it cannot be formed by building blocks.

Before we explain how building blocks form other translation rules, we first address the following question: Why does breadth-first search in word-aligned sentence pairs dictate the formation of building blocks of translation rules in SMT? Other than successful empirical evidence from following this strategy, there is actually no further reasoning. Surely the resulting collection of phrase pairs after breadth-first search is well-ordered; there is no formal (linguistic or mathematical) link between the output of the process of word-aligning sentence pairs and the construction of building blocks of translation rules in SMT. The latter is thus executed axiomatically.

The dominant method that extracts translation rules in SMT is due to Och et al. [109] and Koehn et al. [82]. We refer to this method as the *consistency method.* It encompasses the building blocks described above and unaligned words. In particular, for any word-aligned sentence pair of the bitext, a phrase pair is allowed to become a translation rule if and only if the following conditions hold:

1. There exists at least one word alignment in the phrase pair.

2. If a word in the source phrase is aligned to one or more words in the target sentence, then all such target words must appear in the target phrase. Similarly for the target phrase.

3. The words of the source phrase respect the order of appearance in the source sentence. Similarly for the target phrase.

4. Source phrase consists of words that appear consecutively in the source sentence. Similarly for the target phrase.

Condition 1 forbids the construction of arbitrary translation rules. Condition 2 is an outcome of breadth-first search discussed above. Once the words from breadth-first search are collected, the formation of phrases is not arbitrary: Condition 3 dictates how these words are ordered in the phrase pair. Condition 4 guarantees that a phrase is a substring of the sentence from which it is extracted. We say that a translation rule is a phrase pair that is *consistent with the alignment* of the sentence pair, or consistent for brevity. Figure 1.2 shows an example of a sentence pair with an alignment matrix together with all its consistent phrase pairs. These conditions do not confine the number of words of a consistent phrase pair. Consequently, the largest extracted translation rule consists of as many words as there are in the sentence pair, i.e., it is the sentence pair itself. As mentioned above, these rules are eventually stored in the phrase-table. Since a typical sentence consists of around 20 words, it becomes necessary to limit the length

	t_1	t_2	t_3	t_4	t_5	t_6	t_7
s_1	●			●	●		
s_2		●	●				
s_3					●		
s_4		●	●				
s_5							●

$$P(S,T)= \left\{ \begin{array}{c} (s_5,t_7), \\ (s_1^4,t_1^5),(s_5,t_6^7), \\ (s_1^4,t_1^6), \\ (S,T) \end{array} \right\}$$

Figure 1.2: Left: Sentence pair $(S, T) = (s_1 ... s_5, t_1 ... t_7)$ with an alignment matrix. A dot indicates the existence of alignment between source word s_i and target word t_j. Right: The collection of all consistent phrase pairs. s_1^4 is a shorthand for $s_1 s_2 s_3 s_4$; similarly for all other phrases.

of consistent phrase pairs: A maximum of seven words per phrase in a translation rule is used in practice [82].

Without Condition 4, a phrase pair may have its source phrase, or target phrase, or both to be discontinuous. Discontinuities are replaced with wild card tokens. In this case, in the example of Figure 1.2 a valid discontinuous translation rule is, for instance, $(s_1\ X_1\ s_3,\ t_1\ Y_1\ Y_2\ t_4\ t_5)$. Discontinuous phrase pairs are used in hierarchical SMT [27], although only certain types of discontinuities are allowed, as determined by bilingual parsing. Without using any linguistic information, the appropriate selection of discontinuous phrase pairs and engineering aspects of their inclusion during decoding are matters of ongoing research [54, 56, 70]. In this thesis we primarily deal with continuous translation rules. Nonetheless, all the algorithms, techniques and notions that are developed here can be generalized to the discontinuous case.

1.2 The log-linear model

The phrase-table is formed by all consistent phrase pairs that are extracted from all word-aligned sentence pairs of the bitext. As mentioned above, each entry of this phrase-level dictionary is a translation rule which is equipped with certain features. In Chapter 2 it is explained how these features are computed. What is important here is the fact that the phrase-table completely determines the building blocks that are available to the system to generate a translation. The process of decoding any source sentence S requires solving (1.1). Since the phrase-table is the only tool mapping source phrases to target phrases, sentence S needs to be segmented into phrase-level fragments. Each such source phrase is looked up in the phrase-table and provides candidate target phrases. This is done for all source phrases, which results in several candidate translations for S. The sufficiency $p(T|S)$ of a candidate translation T of S

is computed from a Maximum Entropy function

$$p(T|S) = \frac{1}{Z(S)} \exp\left(\sum_i \lambda_i f_i(S,T) \right), \tag{1.5}$$

where $f_i(S,T)$ is a weight produced by the ith feature of the chosen phrase-table entries and λ_i is the corresponding Lagrange multiplier. Inserting (1.5) in (1.1) we get

$$T^* = \arg\max_T \sum_i \lambda_i f_i(S,T), \tag{1.6}$$

i.e., the partition function $Z(S)$ does not have to be computed. If (S,T) is composed by K phrase-table entries, say $\{(s_1,t_1),...,(s_K,t_K)\}$, then $f_i(S,T)$ is typically given by

$$f_i(S,T) = \sum_{k=1}^{K} \log f_i(s_k,t_k), \tag{1.7}$$

where $f_i(s_k,t_k)$ is the ith feature of (s_k,t_k). There is no definitive value for K and in fact all segmentations of S are equally likely. Also, by taking into account that i) a typical source sentence is composed by approximately 20 words; ii) there are numerous candidate target phrases for a source phrase, it is clearly understood that solving (1.6) based on (1.7) is difficult. Indeed, it can be shown that it is an NP-hard problem [81] with a reduction from the Traveling Salesman problem [162]. The beam search stack decoding algorithm [151, 152] is one of the standard approaches for approximating (1.6) efficiently. The phrase pairs $\{(s_1,t_1^*),...,(s_K,t_K^*)\}$ that eventually form (S,T^*) typically consist of one to three words per source and target phrase, with unigrams being the most common.

Although this description of the SMT framework is certainly incomplete, it is sufficient for posing the research directions of this thesis. It has become apparent that SMT breaks down a complex problem (sentence-level translation) into smaller, manageable tasks (multiple phrase-level translations). The flaw of this framework is that these smaller tasks are rather independent from each other. As mentioned above, the associated features of ordering structure have local impact, i.e., in the vicinity of a couple of words from the position of the candidate target phrase. The possible counterargument of memorizing more translation rules and wider ordering structures leads to artificial but certainly not intelligent approaches. Besides, it has been empirically observed that even in the presence of sufficient training data, the translation of short sentences can be inadequate.

Given source phrase S we attempt to draw insight into the relationship between the translation quality of T^* and the current SMT framework. How does the process of translating S on the phrase-level and with local awareness deter the resulting T^* from being a perfect translation? We have basically already ruled out S's length as well as engineering and hardware constraints as the essential driving forces for translation quality. Less obviously, S's linguistic structural classification is also not of fundamental importance. Surely, adequate translation quality is most likely to be achieved when S is a linguistically simple sentence, i.e., when it consists of subject and predicate only. This holds especially in the case where both the source and target languages are from the same linguistic family. Simple sentences enclose those elements of linearity and independence that are required by the SMT framework to perform well. Successful syntax-based techniques that can accommodate translation for linguistically complex sentences, such as preordering [30, 35, 92], do not necessarily leverage from the SMT framework; patterns between source and target language parse trees, dependency trees etc., could possibly be even more useful, if the SMT framework was more globally-aware.

1.3 The Principle of Compositionality

As mentioned in the beginning of this chapter, an ideal translation of S should preserve S's meaning. We argue that the current SMT framework fails to do so, as it disregards the Principle of Compositionality (PoC). This principle, which is also known as Frege's Principle, is as follows [5]: *The meaning of a complex expression is a function of the meanings of its constituents and the operations performed on those constituents.* None of the key terms 'meaning', 'constituent' and 'operation' have a fixed definition. The PoC has thus been the subject of critique [117], but has also produced fruitful results if interpreted appropriately [68]. For our purposes we have:

- A *constituent* is a word of the complex expression (sentence).
- An *operation* is the act of grouping words in a segment. We assume that each word belongs to a unique segment in a given segmentation. A segmentation is thus a partition of the sentence. Furthermore, we assume that words in a segment are ordered in a way that respects their order of appearance in the sentence. Thus, each segment is a (possibly discontinuous) phrase.
- *Meaning* is viewed as an abstract function M whose domain is phrases, but no further specifications are required. No such computations are carried out explicitly throughout this thesis.

Let $\sigma = \{s_1, s_2, ..., s_K\}$ denote a segmentation of sentence S. According to the PoC, there exists a function $\mathcal{F}$ such that

$$M(S) \;=\; \mathcal{F}\,[\,M(s_1),\; M(s_2),\; ...,\; M(s_K)\,]\,. \tag{1.8}$$

Our key assertion is that the PoC can be satisfied for all possible operations, i.e., for any segmentation of S: If $\sigma' = \{s'_1,\; s'_2, ..., s'_L\}$ is another segmentation of S, then there exists a function $\mathcal{F}'$ such that

$$M(S) \;=\; \mathcal{F}'\,[\,M(s'_1),\; M(s'_2),\; ...,\; M'(s'_L)\,]\,. \tag{1.9}$$

Figure 1.3 shows an example of two segmentations for the sentence `Trade unions have already expressed their concern with respect to the lack of collaborative action.` The segments of σ respect grammar (but not syntax),

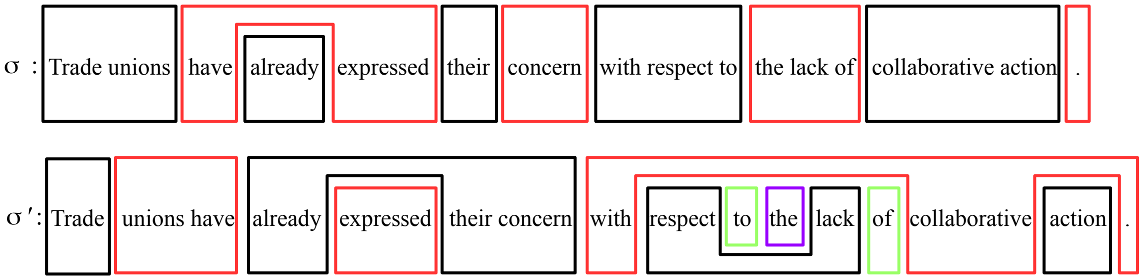


Figure 1.3: Two different segmentations for the same sentence. Boxes separate segments and colors facilitate visualization.

but most importantly they avoid polysemy by having univocal, unambiguous function within the sentence (in fact within *any* sentence). On the other hand, σ' is a rather random segmentation. We assume that the 'effort' required by some $\mathcal{F}$ to process the meanings of σ's segments and produce the meaning of the whole sentence should be less than the 'effort' required by some $\mathcal{F}'$ to perform the same task for σ'. In general, a basic feature that can characterize the degree of 'effort' is whether the function is linear or not. Naturally it is assumed that linearity of $\mathcal{F}$ implies less 'effort'. If $\mathcal{F}$ is linear, then (1.8) is written as

$$M(S) \;=\; M(s_1) \,\oplus\, M(s_2) \,\oplus\, ... \,\oplus\, M(s_K), \tag{1.10}$$

where $\oplus$ is a semantic binary operation.[1] In this thesis we are interested in finding those segmentations of a sentence that correspond to linear functions.

For the pair (S, T^*) composed by the bilingual segmentation $\{(s_1, t_1^*), ..., (s_K, t_K^*)\}$ as generated by (1.6) and (1.7), we ideally expect

$$M(S) \;=\; \mathcal{F}_{S,T^*}\,[\,M(t_1^*),\; M(t_2^*),\; ...,\; M(t_K^*)\,]\,, \tag{1.11}$$

for some function $\mathcal{F}_{S,T^*}$. The subscript of this function stresses that the formation of segments and the combination of their meanings has to take into account the bilingual setting. However, all chosen phrase pairs are equipped with only locally-aware features. Also, the decoding process itself essentially performs independent translations of fragments of S. If we could compute meanings during decoding, the actual resulting meaning would be comparable with

$$\Psi \;=\; M(T_1^*) \,\oplus\, M(T_2^*) \,\oplus\, ... \,\oplus\, M(T_L^*), \tag{1.12}$$

[1] As mentioned above, computations involving meanings are not carried out in this thesis. The details of this binary operator are thus omitted.

where T_l^*'s are possibly overlapping translations of fragments of S. Apart from simple cases, the equality $M(S) = \Psi$ cannot be expected. This thesis attempts to identify under which conditions a bilingual segmentation could validate the statement $M(S) = \Psi$.

To make things clearer, we construct a framework for bilingual segmentations that can potentially adapt to the locally-aware framework of SMT. It is, however, not specifically tailored to SMT and other natural language processing branches, such as composition in distributional semantics [8, 28], could benefit from it. Using the PoC as a starting point, this is achieved by introducing relevant notions and techniques in a monolingual setting, which are then generalized in a bilingual setting via the phrase-table and some empirical results from SMT. We proceed with stating our research questions.

1.4 Research questions

The extraction of translation rules which leads to the formation of the phrase-table with accompanying features has in fact motivated this thesis. Conditions 1–4 above provide an algorithmic tool but no further insights can be drawn.

RQ1 *How can one devise a mathematical framework that is affable to the consistency method? It should be minimal in construction but sufficient for accommodating bilingual segmentations as a generalization. If bilingual segmentations are taken into account, then how do they affect the set of extracted translation rules?* [Chapter 3]

Empirically, it has been shown that only a small proportion of the whole set of translation rules is actually useful during decoding [74, 163]. Within our framework that encompasses bilingual segmentations, we would like to draw further insight into why this is happening, i.e., to identify the qualitative characteristics of the effective subset of translation rules.

A tool that could potentially assist this identification is a phrase-level generative model (in the spirit of the IBM models) that respects consistency. We present such a model but do not carry out the task in full due to heavy engineering difficulties that previous work has pointed out [37]. Instead, a key probability distribution that is encountered in all previous work that constructs phrase-level generative models is explored: The probability of a segmentation of a sentence. In fact, we do something more general with a wider spectrum of applications.

Let S be a finite set and let P_S denote the set of all partitions of S. A probability distribution over P_S is called *exchangeable* if it is invariant under every deterministic rearrangement of places by a permutation of S [57]. For example, if $S = \{1, ..., 6\}$, then an exchangeable distribution p over P_S satisfies, for instance,

$$p(\{\{1,5\},\{2\},\{3,4,6\}\}) = p(\{\{\pi(1),\pi(5)\},\{\pi(2)\},\{\pi(3),\pi(4),\pi(6)\}\}), \quad (1.13)$$

for every permutation π on S. In nonparametric Bayesian statistics, such distributions are traditionally treated as exchangeable; the application of de Finetti's Theorem gives rise to the Dirichlet process [51], the Chinese restaurant process [119], the Pitman-Yor process [118] and the Indian buffet process [60]. For our purposes, however, equalities as in (1.13) are restrictive and undesirable.

RQ2 *How should one construct a method for computing probabilities of non-exchangeable random partitions?* [Chapter 3]

Our search for the said qualitative characteristics essentially begins from a monolingual setting. In particular, we seek conditions that satisfy (1.10).

RQ3 *Given a sentence S in some language, identify what conditions a segmentation $\{s_1, ..., s_K\}$ of S should satisfy in order for (1.10) to hold. How can one define the segmentation that satisfies those conditions optimally?* [Chapter 4]

This optimal segmentation, which we term the *natural segmentation*, is expected to have the same properties as the top segmentation of Figure 1.3 for the corresponding sentence. Its definition is inspired by the construction of measure-theoretic entropy in dynamical systems [77] and builds on paraphrases as in [68]. As such it is difficult to be implemented in practice. To this end, we construct two computationally feasible methods that could possibly simulate natural segmentations. The prominent method builds on a heuristic, the so-called Pointwise Mutual Information (PMI), which has empirically been found to perform well in segmentation-related tasks.

For a given sentence S, in this method we are interested in perturbing a segmentation σ into another segmentation σ' in the following way: Split a single segment of σ into two new segments and form segmentation σ' from these two segments together with the ones that are intact in σ. This operation of 'refinement' along with the set of all possible segmentations of S, forms a particular type of partially ordered set, namely a lattice. We focus on metrics on lattices, because they are interpreted as cost functions for perturbing one segmentation into another.

RQ4 *Given the relationship between Shannon's entropy and metrics on lattices [138], elaborate on the mathematical framework of PMI. Is it possible to extend PMI within this framework for simulating natural segmentations?* [Chapter 4]

The next step is to generalize the definition of natural segmentation from a monolingual to a bilingual setting.

RQ5 *Given a word-aligned sentence pair (S, T), identify what conditions a bilingual segmentation $\{(s_1, t_1), ..., (s_K, t_K)\}$ of (S, T) should satisfy in order to form a bilingual natural segmentation. How can one define the bilingual segmentation that* *satisfies those conditions optimally?* [Chapter 5]

Equipped with a word-aligned bitext that is naturally segmented, we proceed with extracting translation rules.

RQ6 *What is the effect of bilingual natural segmentations on SMT?* [Chapter 5]

The remaining research questions deal with a particular problem in SMT. The phrase-table is the only source of translation rules. A valid translation rule that has not been identified as such during training will not appear in the phrase-table. The construction of new translation rules from existing ones is thus a reasonable research direction. It requires the formation of a new phrase pair and the computation of feature weights for the new pair. It turns out that both tasks are very error-prone and thus hard [21, 114, 166].

Our first attempt to tackle this problem is based on paraphrases, i.e., phrase-level synonymies: If (s, t) is a phrase-table entry and if t' is a paraphrase of t, then (s, t') becomes a new phrase-table entry. We only discuss the first step of this approach, i.e., the identification of t'. We essentially extend the work of Kok and Brockett [87], which harvests paraphrases from the graph representation of a phrase-table It uses a random walk measure, the commute time between two phrases t and t', in order to rank the quality of (t, t').

RQ7 *How should one extend the work of Kok and Brockett [87] in order to identify less noisy pairs of paraphrases and to develop a method that constructs artificial co-occurrence counts for these pairs?* [Chapter 6]

The second attempt suggests a promising strategy for computing ad hoc weights for unseen phrase pairs during decoding. It relies on the compositional nature of phrase pairs, i.e., the building blocks that were discussed in Section 1.1.

The most important weight of a phrase-table entry (s, t) is the translation probability $p(t|s)$ which expresses the likelihood of t being a translation of s in any sentence pair (see Section 2.1 for details). For a pair (s, t) that is not in the phrase-table, i.e., it is unseen, the computation of $p(t|s)$ is a challenge: Not only does it have to reflect the quality of such a translation for s, but its value also has to be in line with the distribution p of the existing phrase-table entries. As mentioned in Section 1.2, the phrase-table is the only tool that maps source phrases to target phrases. Thus, the ability to handle unseen phrase pairs would be useful during decoding.

RQ8 *Given an unseen phrase pair (s, t), how can one compute the translation probability $p(t|s)$ based on their most likely composition of building blocks?* [Chapter 7]

1.5 Thesis overview and origins

This section provides an overview of the thesis and the publications on which each chapter is based. For each publication we mention the role of each co-author.

Chapter 2 is an overview of basic features that are used by most SMT systems. It also provides the baseline SMT system that is used in our experiments in the following chapters. This chapter does not include original material.

Chapter 3 is a rigorous description of the phrase pair extraction mechanism that encompasses bilingual segmentations. It also includes a method for computing probabilities of non-exchangeable random partitions. This chapter is based on *Investigating Connectivity and Consistency Criteria for Phrase Pair Extraction in Statistical Machine Translation* published at MoL'13 by Martzoukos, Costa Florêncio, and Monz [100]. Martzoukos developed the methods. All authors contributed to the text.

Chapter 4 introduces the concept of the natural segmentation of a sentence. It also investigates the relationship between PMI and metrics on lattices that are induced by all segmentations of a sentence. This chapter is partly based on *Maximizing Component Quality in Bilingual Word-Aligned Segmentations* published at EACL'14 by Martzoukos, Costa Florêncio, and Monz [101]. Martzoukos developed the methods and performed the experiments. All authors contributed to the text.

Chapter 5 extends the notion of the natural segmentation to the bilingual level. It also investigates the effect of such segmentations on SMT. This chapter is partly based on the same publication as in Chapter 4.

Chapter 6 describes a graph-based method that harvests paraphrases and assesses the quality of two phrases that are assumed paraphrases of each other. This chapter is based on *Power-Law Distributions for Paraphrases Extracted from Bilingual Corpora* published at EACL'12 by Martzoukos and Monz [99]. Martzoukos developed the methods and performed the experiments. Both authors contributed to the text.

Chapter 7 describes a method for copmuting conditional probabilities of an unseen phrase pair based on alignment information that is provided by its sub-phrase pairs. This chapter is based on unpublished material.

Chapter 8 is a summary of our answers to the research questions that were stated in Section 1.4 and it also includes pointers for future work.

Figure 1.4 shows the dependencies between the chapters of this thesis.

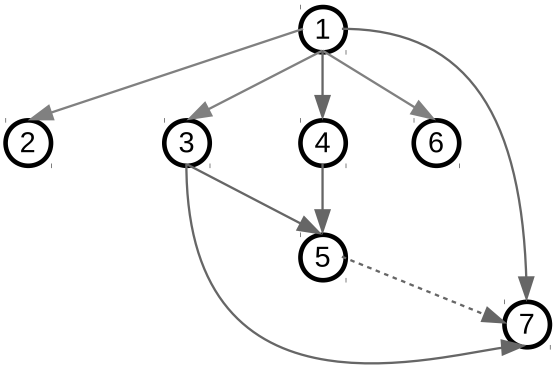


Figure 1.4: Dependency tree for the chapters. Although Chapter 7 is self-contained, its content is best understood if Chapter 3 has also been covered. Further insights about Chapter 7 can be gained if Chapter 5 has been covered (dashed arrow).

Chapter 2

Baseline SMT System

The concepts, techniques and algorithms that are developed in this thesis are integrated in an existing statistical machine translation (SMT) system. In this chapter we outline the components of this system, which is called "Moses" [85] and is open source software.[1] Moses is one of the standard baseline systems that is commonly used in the research literature as well as in WMT, the Workshop on Statistical Machine Translation [15].

2.1 Phrase-table

The phrase-table, which is a phrase-level dictionary, is the focus of this thesis. Each entry is a translation rule, i.e., a phrase pair that has been extracted from aligned parallel corpora after meeting certain criteria. As mentioned in Chapter 1, such a phrase pair must be consistent with the alignment of the sentence pair from which it is extracted. In Chapter 3 we elaborate on what is meant by consistency.

An entry (s, t) is equipped with four features:

- Direct phrase translation probability $p(t|s)$.
- Inverse phrase translation probability $p(s|t)$.
- Direct lexical weighting $lex(t|s)$.
- Inverse lexical weighting $lex(s|t)$.

The direct translation probability is computed as

$$p(t|s) \;=\; \frac{\text{count}(s,t)}{\sum_{t'} \text{count}(s,t')}, \tag{2.1}$$

[1] http://www.statmt.org/moses/

where count(s,t) is the number of times that pair (s,t) has been extracted as a translation rule from the aligned parallel corpus. The inverse translation probability is similarly computed.

Translation probabilities, as given by (2.1), can be unreliable for low-frequency phrase pairs. Lexical weighting assesses the translation quality of pair (s,t) based on correspondences between the pair's most basic constituents, i.e., words. If $(s_1 \ldots s_n, t_1 \ldots t_m)$ denotes a sentence pair from the word-aligned parallel corpus, where each s_i and t_j stands for a source and target word respectively, then consider the alignment matrix

$$A(i,j) = \begin{cases} 1, & \text{if } s_i \text{ and } t_j \text{ are aligned} \\ 0, & \text{otherwise,} \end{cases} \tag{2.2}$$

for all $i = 1,...,n$ and $j = 1,...,m$. Also, if a and x denote a source and target word respectively, then consider the direct word translation probability

$$w(x|a) \;=\; \frac{\text{count}(a,x)}{\sum_{x'} \text{count}(a,x')}, \tag{2.3}$$

where count(a,x) is the number of times that pair (a,x) has been aligned in the parallel corpus. The direct lexical weighting of pair (s,t) is given by

$$lex(t|s) \;=\; \prod_{j=1}^{|t|} \frac{1}{\sum_{i=1}^{|s|} A(i,j)} \;\sum_{i=1}^{|s|} A(i,j)w(t_j|s_i), \tag{2.4}$$

where $|u|$ denotes the number of words in phrase u. Note that this quantity is not a probability as it is not guaranteed that $\sum_t lex(t|s) = 1$. Inverse lexical weighting is similarly computed.

Lexical weightings are heuristics. In Section 3.3 it is explained that translation probabilities are derived via maximum likelihood estimation.

2.2 Language model

During decoding, translation rules are selected from the phrase-table and the formed candidate target strings need to be tested for grammatical cohesion. The purpose of the language model (LM) is to assess the fluency of translation candidates. If V denotes the vocabulary of the target language, then an LM is a function $p : V^N \rightarrow [0,1]$, for some natural number N. For example, if the produced candidate target strings are `the man sleeps` and `the sleeps man`, then an LM should ideally yield $p($`the man sleeps`$) > p($`the sleeps man`$)$.

An LM is constructed from a large monolingual corpus, which may include the target side of the bilingual corpus that is used to train the other models of the SMT system. The dominant approach is to view this corpus as a stochastic process: A corpus W is a sequence of words $w_1, .., w_N$; each word w_i is an instance of random variable, or state, W_i that takes values from the vocabulary V of the target language. Each state W_i is assumed to be dependent only on its $n-1$ preceding states, or history $W_{i-n+1}, ..., W_{i-1}$. If $n = 2$ then W is a Markov Chain. The quantities of interest are the conditional probabilities

$$p_n(w_i \mid w_{i-m+1}^{i-1}) \ := \ p_n(W_i = w_i \mid W_{i-m+1} = w_{i-m+1}, ..., \ W_{i-1} = w_{i-1}), \quad (2.5)$$

for all $1 < m \leq n$ and $1 \leq i \leq N$, wherever histories exist. It is explained in Section 4.2 that for any word w and any sequence of words h, the simplest LM that can assess the quality of string hw is given by

$$p(hw) \equiv p_n(w|h) = \begin{cases} \frac{\text{count}(hw)}{\text{count}(h)}, & \text{if } |h| \leq n-1 \\ \\ \frac{\text{count}(\tilde{h}w)}{\text{count}(\tilde{h})}, & \text{otherwise,} \end{cases} \quad (2.6)$$

where count(x) is the number of times that sequence x has been observed in corpus W, and $\tilde{h} \subset h$ consists of h's final $n-1$ words. The value of n is fixed and the LM is thus referred to as an n-gram LM. The typical choice for n in SMT ranges from three to seven [16, 73, 96, 131]. In this thesis we use 4-gram LMs.

A major drawback of (2.6) is that if sequence h is not observed in corpus W, then $p_n(w|h)$ is undefined. To overcome this problem smoothing is employed, which is a technique for assigning probability mass for unseen events. The smoothing technique of Kneser and Ney [79] interpolates the n-gram LM with lower order m-gram LMs (i.e., $1 \leq m < n$); it has empirically been found to perform well in SMT [18]. Throughout this thesis we thus use 4-gram interpolated LMs with Kneser-Ney smoothing, which are generated by the open source software SRILM.[2] For the details of this LM we refer to [25, 148].

2.3 Reordering model

As with the language model, the reordering model also deals with fluency, but at the bilingual level.

Suppose that word-aligned sentence pairs are read from left to right. The reordering model is a phrase-level dictionary whose entries are exactly the same as the ones in the phrase-table; an entry (s, t) is equipped with features that indicate how likely (s, t) is to be ordered in a sentence pair with respect to the translation rules that precede and follow (s, t).

Figure 2.1 shows how to identify such an order with the aid of an alignment matrix in general. The main rectangle with end points (i, j), $(i, j+m)$, $(i+n, j)$ and $(i+n, j+m)$ stands for a consistent phrase pair, i.e., an extracted translation rule, say $(s, t) = (s_i...s_{i+n}, \ t_j...t_{j+m})$. By definition of consistency (see Chapter 1), there are no

[2] http://www.speech.sri.com/projects/srilm/

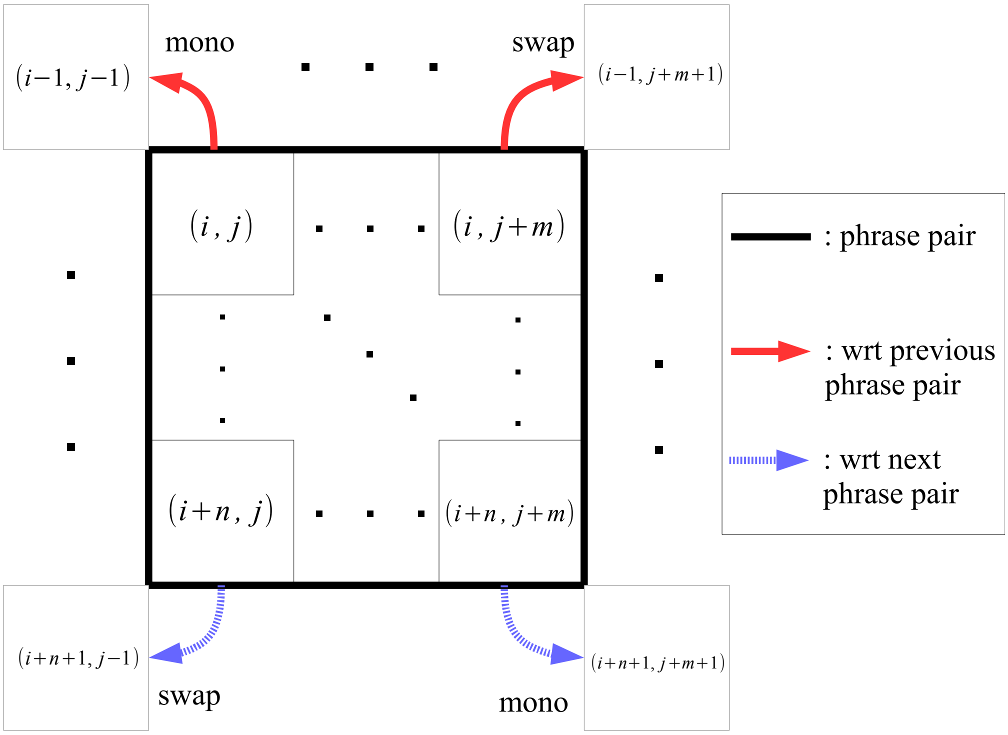


Figure 2.1: All possible continuous orientation types for a consistent phrase pair.

other valid translation rules to the north, south, east nor west of (s, t). Valid translation rules may exist:

- To the northwest or northeast, i.e., immediately to the left of (s, t), and
- To the southwest or southeast, i.e., immediately to the right of (s, t).

For example, if a northwest translation rule (s', t') exists, then the cell $(i-1, j-1)$ should be contained in the rectangle that is formed by (s', t'). Similarly for all other corners of (s, t). Note that translation rules to the northwest and northeast of (s, t) cannot exist simultaneously, because that would violate consistency. What can happen though, is that both cells $(i-1, j-1)$ and $(i-1, j+m+1)$ represent absence of alignment. This case indicates that the translation rule to the left of (s, t) is not immediately to its left, but requires a 'jump' over another translation rule. Similar arguments hold for the south orientation of (s, t).

The resulting orientations are termed monotone, swap and discontinuous. The first two are shown in Figure 2.1 with respect to the immediately previous and immediately next translation rule of (s, t). Let O denote the set of these three orientations. An entry (s, t) of the reordering model is thus equipped with six features:

- Probabilities $p_{\text{prev}}(o|s,t)$, for all $o \in O$. Each represents the likelihood of observing orientation type o with respect to the previous translation rule.
- Probabilities $p_{\text{next}}(o|s,t)$, for all $o \in O$. Each represents the likelihood of observing orientation type o with respect to the next translation rule.

Let count$(o, s, t; \text{prev})$ denote the number of times that orientation type o with respect to the previous translation rule of (s, t) has been observed in the word-aligned parallel corpus. Similarly for count$(o, s, t; \text{next})$. Then the above probabilities are computed as

$$p_z(o|s,t) \;=\; \frac{\text{count}(o, s, t; z)}{\sum_{o' \in O} \text{count}(o', s, t; z)}, \tag{2.7}$$

for all $o \in O$, and for all $z \in \{\text{prev}, \text{next}\}$. In Moses, the above models are smoothed as

$$p_z(o|s,t) \;=\; \frac{1/2 \;+\; \text{count}(o, s, t; z)}{3/2 \;+\; \sum_{o' \in O} \text{count}(o', s, t; z)}, \tag{2.8}$$

for all $o \in O$, and for all $z \in \{\text{prev}, \text{next}\}$.

2.4 Distortion cost

The reordering model operates along with the distortion cost. The latter is not a model that is learned from training data as it operates during decoding only. At this stage, the source sentence is segmented into source phrases.

Suppose that the segmented source sentence is read from left to right. For the leftmost source phrase, a candidate target phrase is sought by looking up the phrase-table. We are not restricted to seeking a translation candidate for the source phrase immediately to the right of the leftmost source phrase. On the contrary, the decoding process allows us to choose which source phrase should be decoded next; we can decode a source phrase that is several phrases to the right, then move again to the left etc. The jumps over source phrases come at a cost, the distortion cost. In Moses, a linear distortion cost is implemented in the decoder and is defined as follows:

Suppose that source sentence S is segmented into K source phrases. Source phrases are decoded one by one and each such phrase is labeled by an integer $k \in \{1, ..., K\}$, which indicates its order in the decoding process. Since each source phrase is decoded to a single target phrase, that target phrase inherits the label. Let

$$\begin{aligned} \text{start}_k \;&= \text{position of the } \textit{first} \text{ word of the source phrase} \\ &\qquad \text{that translates to the } k\text{th target phrase;} \qquad (2.9) \\ \text{end}_k \;&= \text{position of the } \textit{last} \text{ word of the source phrase} \\ &\qquad \text{that translates to the } k\text{th target phrase.} \qquad (2.10) \end{aligned}$$

At the end of the decoding process a target candidate translation T is formed. The distortion cost is given by

$$D(S,T) \;=\; -\sum_{k=1}^{K} d_k, \tag{2.11}$$

with

$$d_k \;=\; |\text{end}_{k-1} + 1 - \text{start}_k|, \tag{2.12}$$

where $\text{end}_0 = -1$. The (per phrase) distortion in (2.12) simply counts the number of words that are jumped in order to proceed to the next phrase that is to be decoded. The distortion cost in (2.11) is the accumulation of these jumps, i.e., the total variation of the decoding scheme.

It has been observed that large d_k results in inadequate translations [59]. Throughout this thesis we limit the (per phrase) distortion to six. This implies that the segmentation of the source sentence and the order with which source phrases are decoded should be such that $d_k \leq 6$, for all k.

2.5 Word and phrase penalty

Word and phrase penalty do not explicitly depend on the output of the training stage. They are (independent) features that control lengths of strings during decoding; their values rely heavily on the resulting Lagrange multipliers of the tuning stage.

Word penalty As mentioned in Section 2.2, a target string $T = t_1...t_N$ that is produced during decoding as a candidate translation of a given source sentence, needs to be assessed for target-language fluency. The LM that performs this task, treats T as a stochastic process with memory n, and the following quantity is computed

$$p(T) \;=\; p(t_1...t_N) \;=\; \prod_{i=1}^{N} p_n(t_i|t_{i-n+1}\;...\;t_{i-1}), \tag{2.13}$$

where probabilities p_n are given by smoothed versions of (2.6). What we need to note in this section is that (2.13) is composed by N probabilities and that it is not normalized for target string length.

A problem arises when we wish to compare the fluency of T with that of another candidate translation $T' = t_1...t_{N'}$ for which $N \neq N'$: Even if T is a more fluent translation than T', it is still more likely that $p(T) < p(T')$, if $N > N'$. The decoder thus favours shorter translation candidates. To counter this bias, the word penalty is used, which is the feature $\omega(T) = e^N$.

Phrase penalty During decoding a source (test) sentence is segmented. These source segments are then looked up in the phrase-table and reordering model and translations are scored accordingly. The phrase penalty controls the size (number of words) of those source segments. Its value is e, i.e., its effect is completely determined by the corresponding Lagrange multiplier.

2.6 Tuning and evaluation

During decoding, a translation candidate is scored by 14 models: Four from the phrase-table (Section 2.1), one from the language model (Section 2.2), six from the reordering model (Section 2.3), the distortion cost (Section 2.4), and two penalties (Section 2.5). Each of these has its corresponding weight, which is a Lagrange multiplier in the log-linear model of (1.5) and (1.7). Tuning is the process of setting values for those weights. This is done with the aid of a development set, or tuning set, which is a parallel corpus different form the one that is used during training; it typically consists of 500–2000 sentence pairs. The process is iterative and works as follows:

1. Set random initial values for the weights.
2. Decode the source side of the tuning set with the existing weights.
3. Compare the resulting translations (output) with the target side of the tuning set (references).
4. Perform local search for new weight values so that the difference between the output and the references is minimal.
5. If the weights do not change much, then stop. Else, go to Step 2.

Local search is performed with Minimum Error Rate Training, or MERT [110]. The metric that is used for comparing strings is the Bilingual Evaluation Understudy, or BLEU [115]. BLEU is a very simple metric that is based on counting which 4-grams of the output also appear in the references.

After the tuning stage is over, the SMT system is ready for usage, i.e., for translating unseen sentences from the source language. If changes or additions have been made to the default models, then testing takes place. This stage requires an additional parallel corpus, the test set, which consists of approximately 2000 sentence pairs [15]; these must not be part of the training nor tuning parallel corpora. The source sentences of the test set are translated by both the default system, or baseline, and by the modified system. Each of the outputs is then compared with the target side of the test set. Again BLEU can be used for this purpose. The comparison of the baseline's BLEU score with the modified system's BLEU score provides feedback on whether the modified system is an adequate SMT system. Throughout this thesis we also use BLEU for scoring the baseline and our system.

Chapter 3

Phrase Pair Extraction

In this chapter we address **RQ1** and **RQ2**. The consistency method, as described in Chapter 1, has been established as the standard strategy for extracting translation rules in statistical machine translation (SMT). However, no attention has been drawn to why this method is successful, other than empirical evidence. Using concepts from graph theory, we identify the relation between consistency and components of graphs that represent word-aligned sentence pairs. It can be shown that translation rules, i.e., phrase pairs of interest to SMT form a sigma-algebra generated by components of such graphs. This construction is generalized by allowing segmented sentence pairs, which a) provides the stepping stone for further understanding of the type of phrase pairs that are useful for SMT, and b) gives rise to a phrase-based generative model. A by-product of b) is a derivation of probability mass functions for random partitions. These are realized as cases of constrained, biased sampling without replacement and we provide an exact formula for the probability of a segmentation of a sentence.

3.1 Introduction

A parallel corpus, i.e., a collection of sentences in a source and a target language, which are translations of each other, is a core ingredient of every SMT system. It serves the purpose of training data, i.e., data from which translation rules are extracted. In its most basic form, SMT does not require the parallel corpus to be annotated with linguistic information, and human supervision is thus restricted to the construction of the parallel corpus.

The extraction of translation rules is done by appropriately collecting statistics from the training data. The pioneering work of Brown et al. [19] identified the minimum assumptions that should be made in order to extract basic translation rules and developed the relevant models that made such extractions possible.

These models, known as IBM models, are based on standard machine learning techniques. Their output is a matrix of word alignments for each sentence pair in the training data. These word alignments provide the input for later approaches that construct phrase-level translation rules which may [156, 159] or may not [95, 109] rely on linguistic information.

The method developed by Och et al. [109], known as the *consistency* method, is a simple yet effective method that has become the standard way of extracting (source, target)-pairs of phrases as translation rules. The development of consistency has been done entirely on empirical evidence and it has thus been termed a heuristic. This brings us to the first research question that is addressed here:

RQ1 *How can one devise a mathematical framework that is affable to the consistency method? It should be minimal in construction but sufficient for accommodating bilingual segmentations as a generalization. If bilingual segmentations are taken into account, then how do they affect the set of extracted translation rules?*

In this chapter it is shown that the method of Och et al. [109] actually encodes a particular type of structural information induced by word alignments. A word-aligned sentence has an obvious graph representation: Each word represents a labeled vertex and an edge between a source type and a target type vertex exists if and only if they are word-aligned. No source-to-source and no target-to-target edges are assumed and the graph is thus bipartite. It is shown that the connected components of such a graph are exactly the building blocks of translation rules in SMT. All further translation rules emerge exactly from the graph union of connected components.

In other words, the phrase-table is constructed entirely from phrase pairs that result from connected components and the graph union of connected components. The phrase-table must be memorized when decoding a source sentence and consequently its number of entries and the length of each entry are forced to meet practical constraints such as runtime behavior and memory limitations. Commonly, the upper bound for the length of a phrase pair is chosen to be 7 words in each side. This empirical choice provides the optimal balance between manageable phrase-table size and translation quality.

However, existing research has shown that translation quality can be preserved by appropriately discarding most of the phrase-table [74]. This is in fact no surprise: By collecting all possible unions of connected components after training, we informally create a huge warehouse of phrase pairs. Such a storage lacks qualitative criteria and decoding is essentially reduced to a smart look-up in the pile of phrase pairs. Such phrase-table filtering methods also use quantitative criteria. Part of this thesis is also devoted to providing further understanding as to what makes this small subset of all possible unions of components important. In this chapter, the stepping stone for the search of qualitative criteria is provided.

This is achieved by allowing source-to-source and target-to-target edges in the graph representation of the word-aligned sentence pair. More precisely same type edges are allowed if and only if the corresponding words are consecutive in the sentence. Thus, segmented sentences are considered and each such segment formalizes the intuitive representation of a phrase. We show that, for fixed word alignments, the set of all phrase pairs of interest to SMT can be traced in the set of connected components of all possible bilingual bilingual segmentations. In other words, the search for qualitative criteria is shifted to understanding what makes such a subset of connected components important.

Although this task is completed in Chapter 5, we describe here another process that could in theory strengthen our thesis points. We outline a generative model in the spirit of the IBM models that finds the most likely connected components from the most likely segmented sentence pairs. Previous work has shown that similar phrase-based generative models provide comparable translation quality with the IBM models. They are, however, harder to train in practice [37, 95].

For a sentence pair (S, T) with $|S|$ and $|T|$ being the number of source and target words, the IBM models seek the most likely alignments between such $|S|$ and $|T|$ words. On the other hand, phrase-based models solve similar maximization problems, but have a much bigger search space: There are $2^{|S|-1}$ possible ways to segment a source sentence and similarly for the target side. The exponential number of possible phrase configurations in a sentence forces one to apply various heuristics and simplifications when Expectation-Maximization is performed on millions of sentence pairs.

As mentioned above, our suggested phrase-based model would provide further evidence about qualitative characteristics of the small subset of all possible phrase pairs that is useful to SMT. However, a potential implementation of our ideas would result in engineering difficulties, as encountered in previous work [37]. This deters us from carrying out this task in full. Nonetheless, we do handle an important quantity quite differently from other research: In all previous work (see Section 3.2) the probability of a segmentation of a sentence requires the explicit modeling of the length of such a segmentation as an additional random variable. We follow a different approach that is closer to coalescent theory. Even within the framework of coalescent theory our approach does not assume that segments in a segmentation are an exchangeable nor a partially exchangeable sequence:

RQ2 *How should one construct a method for computing probabilities of non-exchangeable random segmentations and random partitions in general?*

We thus distance ourselves from the more familiar Chinese Restaurant Process and derive a probability mass function that is closer in construction to the Hyper-Dirichlet type I distribution [43]. This is achieved by considering a sentence segmentation as an outcome of all possible stochastic processes that lead to its formation. This assumption is coupled with a compact graph-based encoding of all possible segmentations of a sentence. Vertices of such graphs are all possible segments of a sentence, each of which has its own weight. These weights are the only parameters of our model. Furthermore, our approach can be trivially extended to modeling random partitions in general.

Another issue that is addressed in this chapter regards the associated translation probabilities of the extracted phrase pairs. It is not difficult to see that all extracted phrase pairs are typically treated as a bag-of-phrase pairs in SMT. Under this assumption, the empirical distribution is the natural choice for translation probabilities. We show that this approach overlooks the richer structure induced by the compositional nature of phrase pairs; the measure space that is formed by the set of all connected components yields promising alternative methods for computing translation probabilities. This realization provides the stepping stone for computing compositionally aware translation probabilities, and is investigated in Chapter 7.

3.2 Related work

To the best of our knowledge, this is the first attempt to investigate formal motivations behind the consistency method.

Several phrase-level generative models have been proposed, almost all relying on multinomial distributions for the phrase alignments [11, 37, 41, 52, 95, 157, 165]. This is a consequence of treating alignments as functions rather than partitions.

Word alignment and phrase extraction via Inversion Transduction Grammars [156], is a linguistically motivated method that relies on simultaneous parsing of source and target sentences [26, 40, 106].

The partition probabilities that will be introduced in Section 3.6.2 share the same tree structure discussed in [43], which has found applications in Information Retrieval [65].

3.3 Definition of consistency

In this section the definition of consistency is presented; it was introduced by Och et al. [109], refined by Koehn et al. [82], and we follow Koehn [86] in our description. We start with some preliminary definitions.

Let $S = s_1 ... s_{|S|}$ be a source sentence, i.e., a string that consists of consecutive source words; each word s_i is drawn from a source language vocabulary and i indicates the position of the word in S. The operation of string *extraction* from the words of S is defined as the construction of the string $s = s_{i_1} ... s_{i_n}$ from the words of S, with $1 \leq i_1 < ... < i_n \leq |S|$. If $i_1, ..., i_n$ are consecutive, which implies that s is a substring of S, then s is called a source phrase and we write $s \subseteq S$. As a shorthand we also write $s_{i_1}^{i_n}$ for the phrase $s_{i_1} ... s_{i_n}$. Similar definitions apply to the target side and we denote by $T,\ t_j$ and t a target sentence, word and phrase respectively.

Let $(S, T) = (s_1 s_2 ... s_{|S|},\ t_1 t_2 ... t_{|T|})$ be a sentence pair and let A denote the $|S| \times |T|$ matrix that encodes the existence/absence of word alignments in (S, T) as

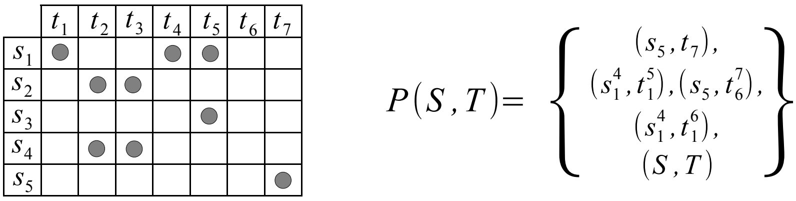
$$A(i, j) = \begin{cases} 1, & \text{if } s_i \text{ and } t_j \text{ are aligned} \\ 0, & \text{otherwise,} \end{cases} \tag{3.1}$$

for all $i = 1, ..., |S|$ and $j = 1, ..., |T|$. Unaligned words are allowed. A pair of strings $(s,t) = (s_{i_1}...s_{i_{|s|}},\ t_{j_1}...t_{j_{|t|}})$ that is extracted from (S,T) is termed *consistent* with A, if the following conditions are satisfied:

1. $s \subseteq S$ and $t \subseteq T$.
2. $\forall k \in \{1, ..., |s|\}$ such that $A(i_k, j) = 1$, then $j \in \{j_1, ..., j_{|t|}\}$.
3. $\forall l \in \{1, ..., |t|\}$ such that $A(i, j_l) = 1$, then $i \in \{i_1, ..., i_{|s|}\}$.
4. $\exists k \in \{1, ..., |s|\}$ and $\exists l \in \{1, ..., |t|\}$ such that $A(i_k, j_l) = 1$.

Condition 1 guarantees that (s,t) is a phrase pair and not just a pair of potentially discontinuous strings. Condition 2 says that if a word in s is aligned to one or more words in T, then all such target words must appear in t. Condition 3 is the equivalent of Condition 2 for the target words. Condition 4 guarantees the existence of at least one word alignment in (s,t).

For a sentence pair (S,T), the set of all consistent phrase pairs with an alignment matrix A is denoted by $P(S,T)$. Figure 3.1 shows an example of a sentence pair with an alignment matrix together with all its consistent phrase pairs.



	t_1	t_2	t_3	t_4	t_5	t_6	t_7
s_1	●			●	●		
s_2		●	●				
s_3					●		
s_4		●	●				
s_5							●

$$P(S,T) = \left\{ \begin{array}{c} (s_5, t_7), \\ (s_1^4, t_1^5), (s_5, t_6^7), \\ (s_1^4, t_1^6), \\ (S,T) \end{array} \right\}$$

Figure 3.1: Left: Sentence pair with an alignment matrix. Dots indicate existence of word alignments. Right: All consistent phrase pairs.

In SMT the extraction of each consistent pair (s,t) from (S,T) is followed by a statistic $f(s,t;S,T)$. Typically $f(s,t;S,T)$ counts the occurrences of (s,t) in (S,T). Any two sentence pairs in the training data are assumed independent, so that the quantity of interest becomes

$$\text{count}(s,t) = \sum_{(S,T)} f(s,t;S,T), \tag{3.2}$$

where the sum is over all sentence pairs in the training data. The set of all consistent pairs is thus treated as a bag of phrase pairs, i.e., a multiset in which each pair (s,t) occurs count(s,t) times.

The next step is to derive translation probabilities $p(t|s)$ and $p(s|t)$ for each pair (s,t) in the multiset. The phrase pairs are assumed to be independent and identically distributed. For the derivation of $p(t|s)$, the quantity of interest is the log-likelihood function

$$\ell \;=\; \sum_{(s,t)} \text{count}(s,t)\;\log p(t|s), \tag{3.3}$$

which should be maximized subject to the constraints $\sum_t \; p(t|s) \;=\; 1$, for all source phrases s in the multiset. The method of Lagrange multipliers yields the solution

$$p(t|s) \;=\; \frac{\text{count}(s,t)}{\sum_{t'} \text{count}(s,t')}, \quad \text{for all } (s,t). \tag{3.4}$$

A similar process is followed in order to estimate $p(s|t)$. Finally, the entries of the phrase-table consist of all extracted phrase pairs, their corresponding translation probabilities and other models that were discussed in Chapter 2.

3.4 Consistency and components

For a given sentence pair (S,T) and a fixed word alignment matrix A, our aim is to show the equivalence between consistency and connectivity properties of the graph formed by (S,T) and A. Moreover, it is explained that the way in which measurements are performed is not compatible, in principle, with the underlying structure. We start with some basic definitions from graph theory; see for example [67].

Let $G\;=\;(V,E)$ be a graph with vertex set V and edge set E. Throughout this chapter, vertices represent words and edges represent word alignments, but the latter will be further generalized in Section 3.5. A *subgraph* $H\;=\;(V',E')$ of G is a graph with $V'\;\subseteq\;V$, $E'\;\subseteq\;E$ and the property that for each edge in E', both its endpoints are in V'. A *path* in G is a sequence of edges which connect a sequence of distinct vertices. Two vertices $u,v\in V$ are called connected if G contains a path from u to v. G is said to be *connected* if every pair of vertices in G is connected.

A connected component, or simply *component*, of G is a maximal connected subgraph of G. Maximal means that it is the largest possible such subgraph: It is impossible to find another vertex or another edge anywhere in G, such that it could be added to the subgraph and the subgraph would still be connected. G is called *bipartite* if V can be partitioned in sets V_S and V_T, such that every edge in E connects a vertex in V_S to one in V_T. The disjoint union of graphs, or simply *union*, is an operation on graphs defined as follows: For n graphs with disjoint vertex sets $V_1,...,V_n$ (and hence disjoint edge sets), their union is the graph $(\bigcup_{i=1}^{n} V_i, \bigcup_{i=1}^{n} E_i)$.

Consider the graph G whose vertices are the words of the source and target sentences, and whose edges are induced by the non-zero entries of the matrix A. There are no edges between any two source-type vertices nor between any two target-type vertices. Moreover, the source and target language vocabularies are assumed to be disjoint and thus G is bipartite. The set of all components of G is defined as C_1. Also, let k denote the cardinality of this set, i.e., $|C_1| = k$. From the members of C_1 sets $C_2, ..., C_k$ are further constructed as follows: For each i, $2 \leq i \leq k$, any member of C_i is formed by the union of any i distinct members of C_1. In other words, any member of C_i is a graph with i components and each such component is a member of C_1. The cardinality of C_i is clearly $\binom{k}{i}$, for every i, $1 \leq i \leq k$.

Note that $C_k = \{G\}$, since G is the union of all members of C_1. Moreover, observe that $C_* = \cup_{i=1}^{k} C_i$ is the set of graphs that can be generated by all possible unions of G's components. In that sense

$$C = \{\emptyset\} \cup C_* \tag{3.5}$$

is the powerset of G. Indeed we have $|C| = 1 + \sum_{i=1}^{k} \binom{k}{i} = 2^k$ as required.[1]

Figure 3.2 shows the graph G and the associated sets C_i of (S, T) and A in Figure 3.1. Note the bijective correspondence between consistent pairs and the phrase pairs that can be extracted from the vertices of the members of the sets C_i. This is a consequence of consistency Conditions 2 and 3, since they provide the sufficient conditions for component formation.

In general, if a pair of strings (s, t) satisfies the consistency Conditions 2 and 3, then it can be extracted from the vertices of a graph in C_i, for some i. Moreover, if Conditions 1 and 4 are also satisfied, i.e., if (s, t) is consistent, then we can write

$$P(S,T) = \bigcup_{i=1}^{k} \left\{ \, (S_H, \, T_H) : \; H \in C_i, \; S_H \subseteq S, \; T_H \subseteq T \, \right\}, \tag{3.6}$$

where S_H denotes the extracted string from the source-type vertices of H, and similarly for T_H. Having established this relationship, when referring to members of C, we henceforth mean either consistent pairs or *inconsistent* pairs. The latter are pairs (S_H, T_H) for some $H \in C$ such that at least either $S_H \nsubseteq S$ or $T_H \nsubseteq T$.

The construction above shows that phrase pairs of interest to SMT are part of a carefully constructed subclass of all possible string pairs that can be extracted from (S, T). The powerset C of G gives rise to a small, possibly minimal, set in which consistent and inconsistent pairs can be *measured*. In order to explain our point some further definitions are needed. The following standard definitions can be found in, e.g., [50].

Let X be a set. A collection B of subsets of X is called a *sigma-algebra* if the following conditions hold:

1. $\emptyset \in B$.

2. If F is in B, then so is its complement $X \setminus F$.

3. If $\{F_i\}$ is a countable collection of sets in B, then so is their union $\bigcup_i F_i$.

[1] Here we used the fact that for any set X with $|X| = n$, the set of all subsets of X, i.e., the powerset of X, has cardinality $\sum_{i=0}^{n} \binom{n}{i} = 2^n$.

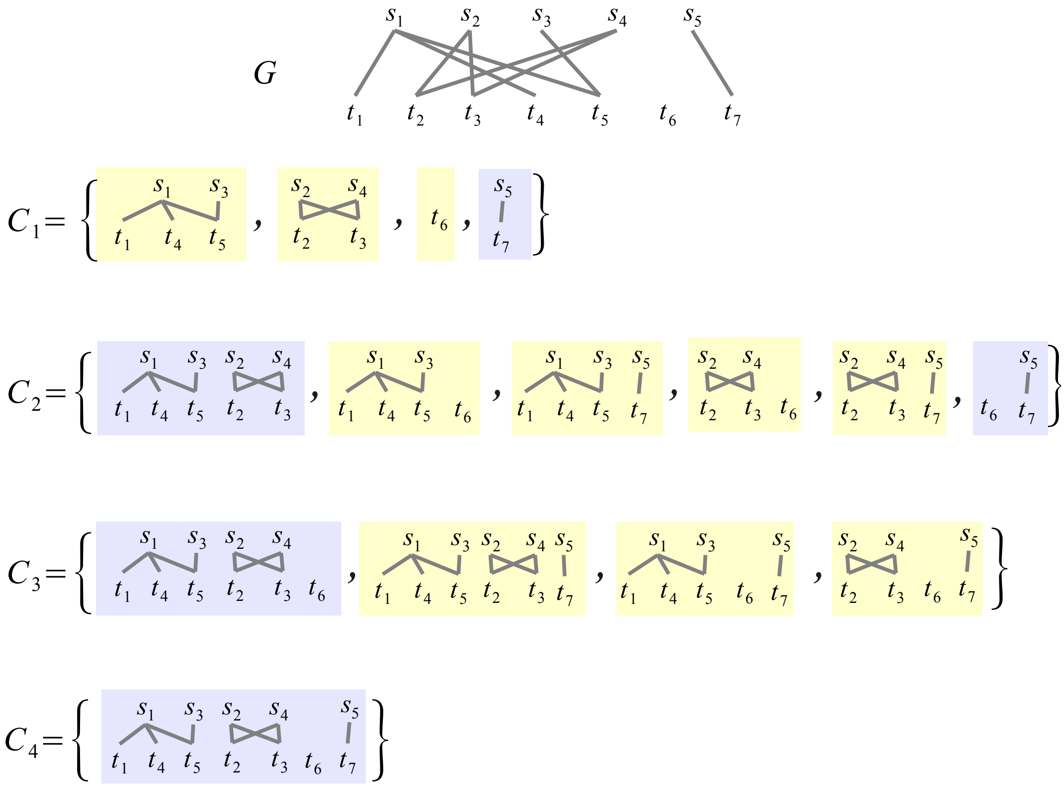


Figure 3.2: The graph representation of the matrix in Figure 3.1, and the sets generated by components of the graph. Dark shading indicates consistency.

Condition 1 guarantees that B is non-empty and Conditions 2 and 3 say that B is closed under complementation and countable unions respectively. The pair (X, B) is called a *measurable space*.

A function $\mu : B \to [0, \infty)$ is called a *measure* if the following conditions hold:

1. $\mu(\emptyset) = 0$.

2. If $\{F_i\}$ is a countable collection of pairwise disjoint sets in B, then

$$\mu\left(\bigcup_i F_i\right) = \sum_i \mu(F_i).$$

Condition 2 is known as *sigma-additivity*. The triple (X, B, μ) is called a *measure space*.

In our case, since C is (by construction) a sigma-algebra, the pair (C_1, C) is a measurable space. Furthermore, one can construct a measure space (C_1, C, f), with an appropriately chosen measure $f : C \to [0, \infty)$.

Is the occurrence-counting measure f of Section 3.3 a good choice? Fix an ordering for C_i, and let $C_{i,j}$ denote the jth member of C_i, for all i, $1 \leq i \leq k$. Furthermore, let $\delta(x, y) = 1$, if $x = y$ and 0, otherwise. We argue by contradiction that the occurrence-counting measure

$$f(H) = \sum_{\{H'\colon\, H' \in C,\; H' \text{ is consistent}\}} \delta(H, H'), \tag{3.7}$$

fails to form a measure space. Suppose that more than one component of G is consistent, i.e., suppose that

$$1 < \sum_{j=1}^{k} f(C_{1,j}) \leq k. \tag{3.8}$$

By construction of C, it is guaranteed that

$$1 = f(G) = f(C_{k,1}) = f\left(\bigcup_{j=1}^{k} C_{1,j}\right). \tag{3.9}$$

The members of C_1 are pairwise disjoint, because each of them is a component of G. Thus, since f is assumed to be a measure, sigma-additivity should be satisfied, i.e., we must have

$$f\left(\bigcup_{j=1}^{k} C_{1,j}\right) = \sum_{j=1}^{k} f(C_{1,j}) > 1, \tag{3.10}$$

which is a contradiction.

Note that (3.7) should not be confused with the standard counting measure. For any measurable space $(X, \mathcal{X})$ with X countable, such a function $f(A)$ equals the number of elements in A, for any finite subset A of $\mathcal{X}$. In our case $f(H)$ would count the number of components in H.

The occurrence-counting measure f of Section 3.3 operates in a different measure space, one that overlooks the compositional nature of extracted phrase pairs. All extracted phrase pairs are treated as a bag-of-phrase pairs and the occurence counting measure is, naturally, the empirical measure. In Chapter 7 we explain how to combine (C_1, C) with the bag-of-phrase pairs in order to derive a statistically elaborate relationship between phrase pairs and their components.

3.5 Consistency, components and segmentations

In Section 3.4 the only relation that was assumed among source (target) words/vertices was the order of appearance in the source (target) sentence. As a result, the graph representation G of (S, T) and A was bipartite. There are several, linguistically motivated, ways in which a general graph can be obtained from the bipartite graph G. We explain that the minimal linguistic structure, namely sentence segmentations, can provide a generalization of the construction introduced in Section 3.4.

Let X be a finite set of consecutive integers. A *consecutive partition* of X is a partition of X such that each part consists of integers consecutive in X. A *segmentation* σ of a source sentence S is a consecutive partition of $\{1, ..., |S|\}$. A part of σ, i.e., a segment, is intuitively interpreted as a phrase in S. In the graph representation G of (S, T) and A, a segmentation σ of S is realized by the existence of edges between consecutive source-type vertices whose labels, i.e., word positions in S, appear in the same segment of σ. The same argument holds for a target sentence and its words; a target segmentation is denoted by τ.

Clearly, there are $2^{|S|-1}$ possible ways to segment S and, given a fixed alignment matrix A, the number of all possible graphs that can be constructed is thus $2^{|S|+|T|-2}$. The bipartite graph of Section 3.4 is just one possible configuration, namely the one in which each segment of σ consists of exactly one word, and similarly for τ. This segmentation pair is denoted by (σ_0, τ_0).

We now turn to extracting consistent pairs in this general setting from all possible segmentations (σ, τ) for a sentence pair (S, T) and a fixed alignment matrix A. As in Section 3.4, we construct graphs $G^{\sigma,\tau}$, associated sets $C_i^{\sigma,\tau}$, for all i, $1 \leq i \leq k^{\sigma,\tau}$, and $C^{\sigma,\tau}$, for all (σ, τ). Consistent pairs are extracted in lieu of (3.6), i.e.,

$$P^{\sigma,\tau}(S,T) = \bigcup_{i=1}^{k^{\sigma,\tau}} \{\ (S_H,\ T_H) :\ H \in C_i^{\sigma,\tau},\ S_H \subseteq S,\ T_H \subseteq T\ \}, \tag{3.11}$$

and it is trivial to see that

$$\{(S,T)\} \subseteq P^{\sigma,\tau}(S,T) \subseteq P(S,T), \tag{3.12}$$

for all (σ, τ). Note that $P(S,T) = P^{\sigma_0,\tau_0}(S,T)$ and, depending on the details of A, it is possible for other pairs (σ, τ) to attain equality. Moreover, each consistent pair in $P(S,T)$ can be be extracted from a member of at least one $C^{\sigma,\tau}$.

We focus on the sets $C_1^{\sigma,\tau}$, i.e., the components of $G^{\sigma,\tau}$, for all (σ, τ). In particular, we are interested in the relation between $P(S,T)$ and $C_1^{\sigma,\tau}$, for all (σ, τ). Each consistent $H \in C^{\sigma_0,\tau_0}$ can be converted into a single component by appropriately forming edges between consecutive source-type vertices and/or between consecutive target-type vertices. The resulting component will evidently be a member of $C_1^{\sigma,\tau}$, for some (σ, τ). It is important to note that the conversion of a consistent $H \in C^{\sigma_0,\tau_0}$ into a single component need not be unique; see Figure 3.3 for a counterexample. Since (a) such conversions are possible for all consistent $H \in C^{\sigma_0,\tau_0}$ and (b) $P(S,T) = P^{\sigma_0,\tau_0}(S,T)$, it can be deduced that all possible consistent pairs can be traced in the sets $C_1^{\sigma,\tau}$, for all (σ, τ). In other words, we have:

$$P(S,T) = \bigcup_{\sigma,\tau} \{\ (S_H,\ T_H) :\ H \in C_1^{\sigma,\tau},\ S_H \subseteq S,\ T_H \subseteq T\ \}. \tag{3.13}$$

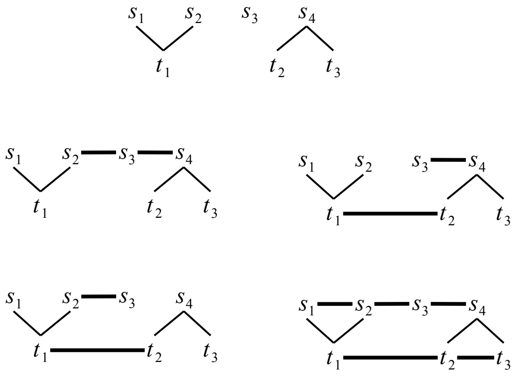


Figure 3.3: A graph with three components (top), and four possible conversions into a single component by forming edges between contiguous words.

The above equation says that by taking sentence segmentations into account, we can recover all possible consistent pairs, by inspecting only the components of the underlying graphs.

It would be interesting to investigate the relation between measure spaces $(C_1^{\sigma,\tau}, C^{\sigma,\tau}, f^{\sigma,\tau})$ and different configurations for A. We leave that for future work and focus on the advantages provided by (3.13). In Chapter 5 we show how to extract a subset of $P(S,T)$ using (3.13), while providing the qualitative characteristics of this subset. Such a subset will be shown to be the sufficient set of phrase pairs in SMT. In the next section it is explained how to construct a phrase-based generative model that finds this subset of $P(S,T)$.

3.6 Towards a phrase-level model that respects consistency

The aim of this section is to exploit the relation established in (3.13) between consistent pairs and components of segmented sentence pairs. We present a phrase-based generative model in the spirit of the IBM models that extracts the most likely consistent phrase pairs from the most likely segmented sentence pairs.

Previous work [37, 39] has highlighted the difficulty of implementing such models. As the resulting phrase-tables do not provide serious added value from the ones generated by the IBM models, our attempt is purely for demonstration.

Our actual contribution is to derive an exact probability mass function for random segmentations. The proposed model is fundamentally different from previous work in SMT. In particular, the probability of a segmentation of a sentence does not require the length of the segmentation to be explicitly modeled as an additional random variable. We follow a different approach, one that is closer to coalescent theory [10]; even within this framework our approach does not assume that segments in a segmentation are an exchangeable nor a partially exchangeable sequence.

3.6.1 Hidden variables

The segmentation of a sentence is a partition of the sentence whose parts are phrases. The formation of components in a sentence pair admits a partition of the sentence pair whose parts are phrase pairs coupled with word alignments. In a segmented sentence pair (σ, τ), the identification of components $C_1^{\sigma,\tau}$ is the key aspect that defines consisntency, as (3.13) suggests. In this section, for a given sentence pair, we are interested in inferring the most likely consistent phrase pairs from the most likely segmented sentence pair. The expilicit identification of word alignments that forms each component in $C_1^{\sigma,\tau}$ is not sought. To put it diffrently, we seek the most likely formation of $C_1^{\sigma,\tau}$ without explicit word alignment information. This quantity is a partition of a segmented sentence pair whose parts are consistent phrase pairs. Such a set of phrase pairs is refferred to as a *bisegmentation* and is denoted by K. Thus, our aim is to identify the most likely K for the most likely (σ, τ) of a sentence pair.

Figure 3.4 shows three possible bisegmentations for sentence pair (s_1^4, t_1^6) with $\sigma = \{\{1,2\},\{3\},\{4\}\} \equiv \{x_1, x_2, x_3\}$ and $\tau = \{\{1\},\{2,3,4\},\{5\},\{6\}\} \equiv \{y_1, y_2, y_3, y_4\}$. In each case the exact alignment information is unknown and we have:

$$\text{(a) } K = \Big\{ \; (x_1,\; y_1),\; (x_2,\; \emptyset),\; (\emptyset,\; y_2),\; (x_3,\; y_3y_4) \; \Big\};$$

$$\text{(b) } K = \Big\{ \; (x_1x_2,\; y_1y_2y_3),\; (x_3,\; y_4) \; \Big\};$$

$$\text{(c) } K = \Big\{ \; (x_1,\; y_3),\; (x_2x_3,\; y_1y_2),\; (\emptyset,\; y_4) \; \Big\}.$$

In the proposed phrase-level generative model the random variables whose instances are σ, τ and K are hidden variables. As with the IBM models, they are associated with the positions of words in a sentence, rather than the words themselves. Alignment information is implicitly identified via the bisegmentation K.

Suppose we have a corpus that consists of pairs of parallel sentences, and let $f_{S,T}$ denote the occurrence count of (S, T) in the corpus. Also, let $l_S = |S|$ and $l_T = |T|$. The aim is to maximize the corpus log-likelihood function

$$\begin{aligned} \ell &= \sum_{S,T} f_{S,T} \;\log p_\theta(T|S) \\ &= \sum_{S,T} f_{S,T} \;\log \sum_{\sigma,\tau,K} p_\theta(T, \sigma, \tau, K|S), \end{aligned} \tag{3.14}$$

where σ, τ and K are hidden variables parameterized by a vector θ of unknown weights, whose values are to be determined. The expectation maximization algorithm [36] sug-

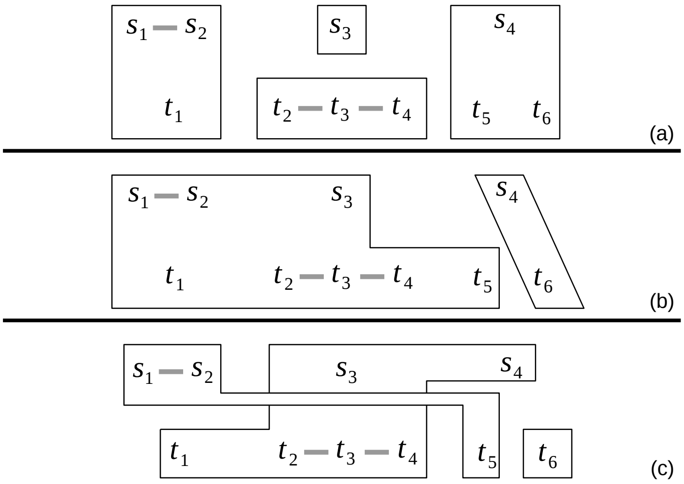


Figure 3.4: Three possible ways to construct bisegmentations for (s_1^4, t_1^6) and a given segmentation pair. Exact word alignment information is unknown. Each polygon represents a part in the bisegmentation.

gests that an iterative application of

$$\theta_{n+1} \;=\; \arg\max_{\theta} \sum_{S,T} f_{S,T} \sum_{\sigma,\tau,K} p_{\theta_n}(\sigma,\tau,K|S,T)\; \log p_\theta(T,\sigma,\tau,K|S), \tag{3.15}$$

provides a good approximation for the maximum value of ℓ. As with the IBM models we seek probability mass functions (PMFs) of the form

$$p_\theta(T,\sigma,\tau,K|S) \;=\; p_\theta(l_T|S)\, p_\theta(\sigma,\tau,K|l_T,S)\, p_\theta(T|\sigma,\tau,K,l_T,S), \tag{3.16}$$

and decompose further as

$$p_\theta(\sigma,\tau,K|l_T,S) \;=\; p_\theta(\sigma,\tau|l_T,S)\, p_\theta(K|\sigma,\tau,l_T,S) \tag{3.17}$$

A further simplification of $p_\theta(\sigma,\tau|l_T,S) = p_\theta(\sigma|S)p_\theta(\tau|l_T)$ may not be desirable, but will help us understand the relation between θ and the PMFs. In particular, we give a formal description of $p_\theta(\sigma|S)$ and then explain that $p_\theta(K|\sigma,\tau,l_T,S)$ and $p_\theta(T|\sigma,$ $\tau,K,l_T,S)$ can be computed in a similar way.

3.6.2 Constrained, biased sampling without replacement

The probability of a segmentation given a sentence can be realised in two ways. We first provide a descriptive approach which is more intuitive, and we use the sentence $S = s_1^4$ as an example whenever necessary. The set of all possible segments of S is denoted by $seg(S)$ and trivially $|seg(S)| = |S|\big(|S|+1\big)/2$. Each segment $x \in seg(S)$ has a nonnegative weight $\theta(x|l_S)$ such that

$$\sum_{x \in seg(S)} \theta(x|l_S) = 1. \tag{3.18}$$

Suppose we have an urn that consists of $|seg(S)|$ weighted balls; each ball corresponds to a segment of S. We sample without replacement with the aim of collecting enough balls to form a segmentation of S. When drawing a ball x we simultaneously remove from the urn all other balls x' such that $x \cap x' \neq \emptyset$. We stop when the urn is empty. In our example, let the urn contain 10 balls and suppose that the first draw is $\{1, 2\}$. In the next draw, we have to choose from $\{3\}$, $\{4\}$ and $\{3, 4\}$ only, since all other balls contain a '1' and/or a '2' and are thus removed. The sequence of draws that leads to a segmentation is thus a path in a decision tree. Since σ is a set, there are $|\sigma|!$ different paths that lead to its formation. The set of all possible segmentations, in all possible ways that each segmentation can be formed, is encoded by the collection of all such decision trees.

The second realization, which is based on the notions of cliques and neighborhoods, is more constructive and will give rise to the desired PMF. A *clique* in a graph is a subset U of the vertex set such that for every two vertices $u, v \in U$, there exists an edge connecting u and v. For any vertex u in a graph, the *neighborhood* of u is defined as the set $N(u) = \{v : \{u, v\} \text{ is an edge}\}$. A *maximal clique* is a clique U that is not a subset of a larger clique: For each $u \in U$ and for each $v \in N(u)$ the set $U \cup \{v\}$ is not a clique.

Let $\mathcal{G}$ be the graph whose vertices are all segments of S and whose edges satisfy the condition that any two vertices x and x' form an edge if and only if $x \cap x' = \emptyset$; see Figure 3.5 for an example. $\mathcal{G}$ essentially provides a compact representation of the decision trees discussed above.

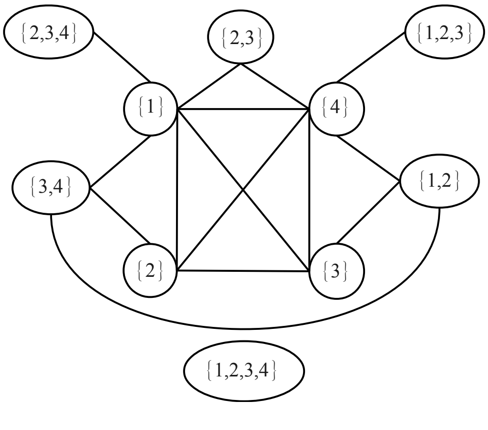


Figure 3.5: The graph whose vertices are the segments of s_1^4 and whose edges are formed by non-overlapping vertices.

It is not difficult to see that a maximal clique also forms a segmentation. Moreover, the set of all maximal cliques in $\mathcal{G}$ is exactly the set of all possible segmentations for S. Thus, $p_\theta(\sigma|S)$ should satisfy

$$p_\theta(\sigma|S) = 0, \text{ if } \sigma \text{ is not a clique in } \mathcal{G}, \tag{3.19}$$

and

$$\sum_\sigma p_\theta(\sigma|S) = 1, \tag{3.20}$$

where the sum is over all maximal cliques in $\mathcal{G}$. In our example $p_\theta\big(\{\ \{1\}, \{1,2\}\ \}|S\big) = 0$, because there is no edge connecting segments $\{1\}$ and $\{1,2\}$ so they are not part of any clique.

In order to derive an explicit formula for $p_\theta(\sigma|S)$ we focus on a particular type of paths in $\mathcal{G}$. A path is called *clique-preserving*, if every vertex in the path belongs to the same clique. Our construction should be such that each clique-preserving path has positive probability of occurring, and all other paths should have probability 0. We proceed with calculating probabilities of clique-preserving paths based on the structure of $\mathcal{G}$ and the constraint of (3.18).

The probability $p_\theta(\sigma|S)$ can be viewed as the probability of generating all clique-preserving paths on the maximal clique σ in $\mathcal{G}$. Since σ is a clique, there are $|\sigma|!$ possible paths that span its vertices. Let $\sigma = \{x_1, ..., x_{|\sigma|}\}$, and let π denote a permutation of $\{1, ..., |\sigma|\}$. We are interested in computing the probability $q_\theta(x_{\pi(1)}, ..., x_{\pi(|\sigma|)})$ of generating a clique-preserving path $x_{\pi(1)}, ..., x_{\pi(|\sigma|)}$ in $\mathcal{G}$. Thus,

$$\begin{aligned} p_\theta(\sigma|S) &= p_\theta(\{x_1, ..., x_{|\sigma|}\}|S) \\ &= \sum_\pi q_\theta(x_{\pi(1)}, ..., x_{\pi(|\sigma|)}) \\ &= \sum_\pi q_\theta(x_{\pi(1)})\ q_\theta(x_{\pi(2)}|x_{\pi(1)}) \times ... \\ &\quad ... \times q_\theta(x_{\pi(|\sigma|)}|x_{\pi(1)}, ..., x_{\pi(|\sigma|-1)}). \end{aligned} \tag{3.21}$$

The probabilities $q_\theta(\cdot)$ can be explicitly calculated by taking into account the following observation: A clique-preserving path on a clique σ can be realised as a sequence of vertices $x_{\pi(1)}, ..., x_{\pi(i)}, ..., x_{\pi(|\sigma|)}$ with the following constraint: If at step $i-1$ of the path we are at vertex $x_{\pi(i-1)}$, then the next vertex $x_{\pi(i)}$ should be a neighbor of all of $x_{\pi(1)}, ..., x_{\pi(i-1)}$. In other words we must have

$$x_{\pi(i)} \in N_{\pi,i} \equiv \bigcap_{l=1}^{i-1} N(x_{\pi(l)}). \tag{3.22}$$

Thus, the probability of choosing $x_{\pi(i)}$ as the next vertex of the path is given by

$$q_\theta(x_{\pi(i)}|x_{\pi(1)}, ..., x_{\pi(i-1)}) = \frac{\theta(x_{\pi(i)}|l_S)}{\sum\limits_{x \in N_{\pi,i}} \theta(x|l_S)}, \tag{3.23}$$

if $x_{\pi(i)} \in N_{\pi,i}$ and 0, otherwise. When choosing the first vertex of the path (the root in the decision tree) we have $N_{\pi,1} = seg(S)$, which gives $q_\theta(x_{\pi(1)}) = \theta(x_{\pi(1)}|l_S)$, as required. Therefore (3.21) can be written compactly as

$$p_\theta(\sigma|S) = \left(\prod_{i=1}^{|\sigma|} \theta(x_i|l_S)\right) \sum_\pi \frac{1}{Q_\theta(\sigma, \pi; S)}, \tag{3.24}$$

where

$$Q_\theta(\sigma, \pi; S) = \prod_{i=1}^{|\sigma|} \sum_{x \in N_{\pi,i}} \theta(x|l_S) \ . \tag{3.25}$$

The construction above can be generalized in order to derive a PMF for any random variable whose values are partitions of a set. Indeed, by allowing the vertices of $\mathcal{G}$ to be a subset of a powerset, and keeping the condition of edge formation the same, probabilities of clique-preserving paths can be calculated in the same way. Figure 3.6 shows the graph $\mathcal{G}$ that represents all possible instances of K with $(S, T) = (s_1^4, t_1^5)$, $\sigma = \big\{\{1, 2\}, \{3\}, \{4\}\big\}$ and $\tau = \big\{\{1\}, \{2, 3, 4\}, \{5\}\big\}$. Again each maximal clique is a possible bisegmentation.

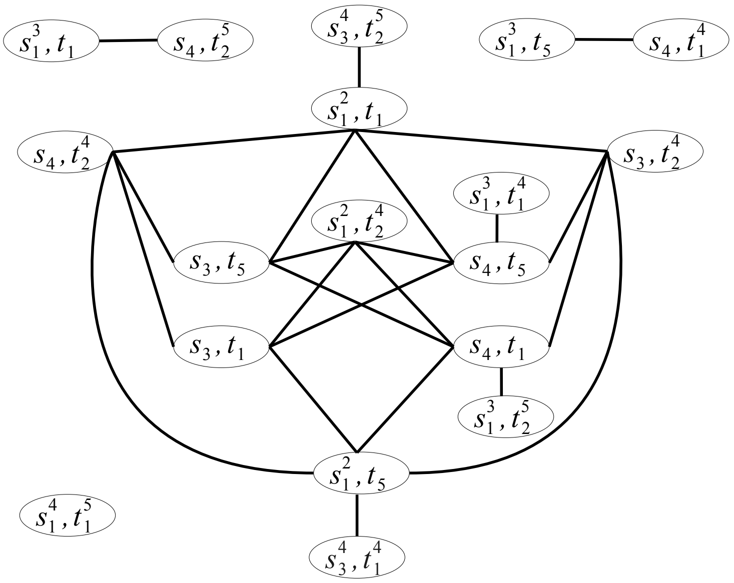


Figure 3.6: Similar to Figure 3.5 but for bisegmentations with $(S, T) = (s_1^4, t_1^5)$ and a given segmentation pair (see text). For clarity, we show the phrases that are formed from joining contiguous segments in each pair, rather than the segments themselves. Also for clarity, unaligned segments are not shown.

In order for this model to be complete, one should solve the maximization step of (3.15) and calculate the posterior $p_{\theta_n}(\sigma, \tau, K|S, T)$. We are not bereft of hope, as relevant techniques have been developed (see Section 3.2).

3.7 Discontinuous phrase pairs

In Sections 3.3–3.5 all consistent phrase pairs (s, t) satisfy the Condition $s \subseteq S$ and $t \subseteq T$, for a given sentence pair (S, T). From a linguistic point of view, this is a purely artificial condition. It does, however, constrain the total number of extracted translation rules. Additionally, the potential inclusion of inconsistent phrase pairs in the phrase-table makes decoding much harder. Such phrase pairs encode their discontinuities with a wild card token. In the example of Figures 3.1 and 3.2 we have for instance $(s_1 X s_3,\ t_1 X X t_4 t_5)$ as an extracted phrase pair from the set of components C_1.

Such phrase pairs are referred to in the literature as *discontinuous* or *gappy* phrase pairs [7, 33, 54, 56, 136]. We primarily deal with continuous phrase pairs and discontinuities are further discussed in Chapter 7.

The inclusion of discontinuous phrases does not change the construction of the PMFs for random partitions in Section 3.6.2. Figures 3.7 and 3.8 show the augmented graphs of Figures 3.5 and 3.6 respectively. The augmentation happens due to the inclusion of vertices that represent discontinuous phrases (Figure 3.7) and discontinuous phrase pairs (Figure 3.8).

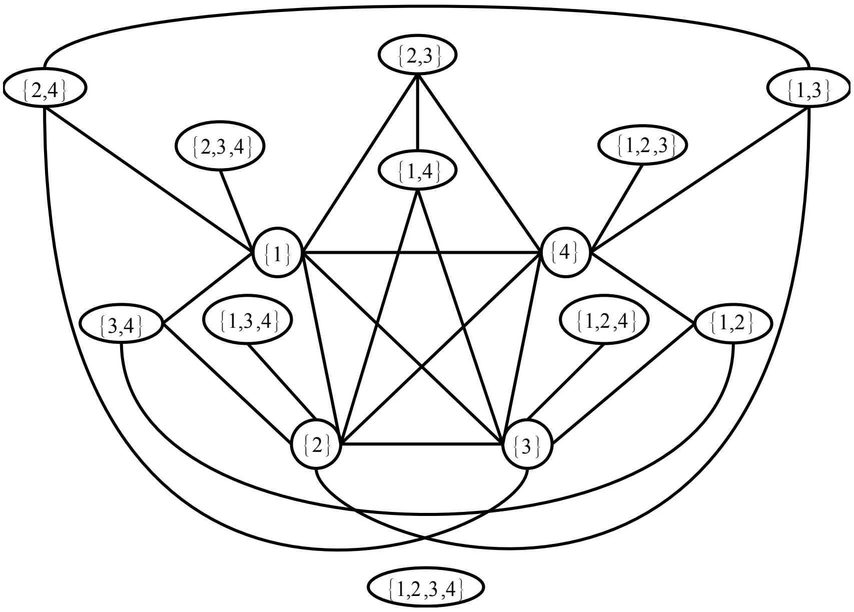


Figure 3.7: Same as Figure 3.5 but with discontinuous segments included.

3.8 Conclusions

In this chapter we presented a formal treatment of the dominant strategy that generates translation rules in SMT, namely consistent phrase pairs. Prior to our work, this

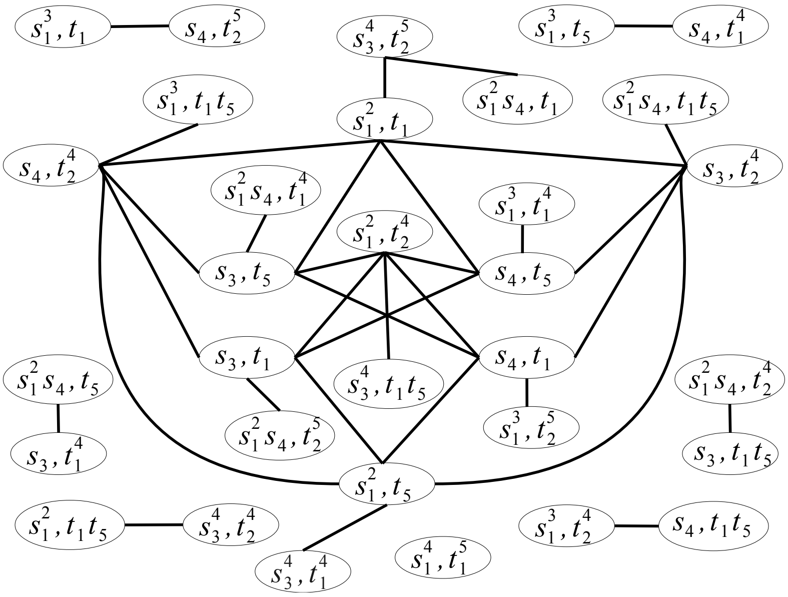


Figure 3.8: Same as Figure 3.6 but with discontinuous components included.

strategy was termed a heuristic and its dominance was entirely based on successful experimental evidence. Our aim was to devise a mathematical framework that is affable to this strategy; it should be able to articulate the links (if any) between the output of word-aligned sentence pairs and qualitative criteria of consistent phrase pairs. Indeed our construction provided insight which gave rise to further exploration, as discussed in the following chapters. The following research question was addressed in this chapter:

RQ1 *How can one devise a mathematical framework that is affable to the consistency method? It should be minimal in construction but sufficient for accommodating bilingual segmentations as a generalization. If bilingual segmentations are taken into account, then how do they affect the set of extracted translation rules?*

First, it was explained that a word-aligned sentence pair has a graph representation as follows: Its source and target language words can be viewed as source and target type vertices respectively; word alignments play the role of edges that connect source and target type vertices. Such a graph is bipartite because no source-to-source nor target-to-target edges are assumed. Word alignments admit a natural partition for this graph: Each part, or component, consists of words from the sentence pair and alignments that connect source words with target words in the following way: a) it is possible to form a path between any two words of the component via word alignments, and b) it is impossible to form such a path between any word in the component and any word outside the component. As mentioned above, for a given word-aligned sentence pair, empirical evidence dictates that only certain types of phrase pairs generate translation rules. We established that a phrase pair is a translation rule if and only if the following conditions hold: i) its words respect the order of appearance in the sentence pair. ii) its words are in one-to-one correspondence with the words of *a union of components of the bipartite graph*. Equivalently, a translation rule is formed by taking an arbitrary collection of components, extracting its words and then ordering them in a way so that the resulting phrase pair is a substring of the sentence pair.

Second, it was explained that the graph representation of a word-aligned sentence pair is just one possible configuration of a more general system, namely the one that allows consecutive words in a sentence to be connected via edges, for both sentences. Under this generalized system, it was shown that the same set of translation rules can be extracted in a different way: A phrase pair is a translation rule if and only if i) its words respect the order of appearance in the sentence pair. ii) its words are in one-to-one correspondence with the words of *a component of a configuration*. This implies that all translation rules can be recovered by collecting components (and components only, not arbitrary unions thereof) from all possible configurations.

The above result lead to the formation of a phrase-based generative model. In the spirit of IBM models, this model extracted the most likely translation rules from the most likely segmentation of sentences of a given sentence pair. Similar previous work has highlighted engineering difficulties during implementation as well as debatable translation quality. Thus, our attempt was purely demonstrative. However, a probability mass function that was present in our model, and prevalent in similar phrase-based models, was treated from a fundamentally different perspective:

RQ2 *How should one construct a method for computing probabilities of non-exchangeable random segmentations and random partitions in general?*

The probability of a segmentation of a sentence was viewed as a case of constrained, biased sampling without replacement. We thus derived a probability mass function that is closer in construction to the Hyper-Dirichlet type I distribution rather than the more familiar, but less applicable, Chinese Restaurant Process. This was achieved by considering a sentence segmentation as an outcome of all possible stochastic processes that lead to its formation. This assumption was coupled with a compact graph-based encoding of all possible segmentations of a sentence. Furthermore, our construction can be trivially extended to modeling non-exchangeable random partitions in general.

Based on the findings of this chapter, we show in Chapter 5 how to extract a particular type of bilingual segmentation of an aligned sentence pair, namely the bilingual natural segmentation, and we investigate the effect of such a component structure on SMT.

Chapter 4

Monolingual Segmentations

In this chapter we address **RQ3** and **RQ4**. The aim of this chapter is to identify what constitutes an ideal segmentation of a sentence in any language. In a bilingual setting, i.e., for a pair of sentences that are translations of each other, the equivalent 'ideal' segmentation is traced in the set of components, as generated by alignments. In a monolingual setting, this ideal segmentation, termed the natural segmentation, should satisfy the following conditions: 1) Substitution of its segments with their paraphrases, yields new sentences that do not deviate much semantically from the original sentence. 2) Segments meeting Condition 1) should be minimal. The formal definition of natural segmentation that is presented, stems from the definition of entropy in dynamical system. As such, it is not easily amenable to hands-on applications, at least within the scope of this thesis. To this end, two novel segmentation methods are developed, with the scope of simulating the output of a natural segmentation. The first one is a generalization of usual n-gram Language Models. It challenges the notion of fixed memory in stochastic processes in a very straight-forward way. Although it is of theoretical interest with many potential applications, its output will not be shown to simulate the desired one. The second method is based on appropriately choosing metrics on lattices, a particular type of partially ordered sets. For our purposes, the set of all segmentations of a sentence, together with the operation of refinement forms a lattice. This method is also of theoretical interest as it demonstrates previously unknown origins of the so-called Pointwise Mutual Information. Additionally, experiments provide evidence for its compatibility with natural segmentations.

4.1 Introduction

In Chapter 3 we showed that the set of translation rules that are useful to Statistical Machine Translation (SMT) emerges from the set of components that is formed from all aligned bilingual segmentations of a parallel corpus. It is also well-known that only approximately 10% of all such translation rules have an actual impact during decoding [74, 163]. In our search for qualitative characteristics of this small subset we turn to a monolingual setting. In order to motivate this shift we provide the conjectured intuitive interpretation of phrase pairs that emerge from components.

Inspection of word alignments as well as the remaining phrase table after pruning results in the following observation: Word alignments capture maximal bilingual associations within minimal (in terms of size) bilingual chunks. In the absence of sentence segmentations, components give rise to phrase pairs which are intuitively interpreted as a subset of bilingual building blocks; the remaining such building blocks are given by some unions of components.

By "bilingual building block" we vaguely mean a phrase pair that captures minimal correspondences between collocations, idioms, multiword expressions and/or fundamental grammatical characteristics. In the presence of sentence segmentations, the small subset of phrase pairs that is useful to SMT is entirely constructed from components. Since these phrase pairs are the same as the ones in the unsegmented case, we focus on identifying what makes components of certain bilingual aligned segmentations interesting. As such components are assumed to be the bilingual building blocks, it is natural to ask whether the involved monolingual segments are the building blocks in their respective languages.

But what does "monolingual building block" mean? It is possible to provide a definition depending on the branch of Computational Linguistics; a universal definition appears to be indifferent. We will attempt to provide a monolingual definition for building blocks that allows to be generalized in a multilingual setting. In other words, we first explain which segments are the building blocks of a sentence and then show that, in a bilingual aligned settings, these segments are generalized by components. We do not claim that monolingual building blocks should be constructed in such a way so that their generalizations are the ones that are useful to SMT. On the contrary, we treat the problem of segmenting sentences as an independent one, seeking the supposedly indifferent universal qualitative criteria for monolingual building blocks. Thus, in this chapter we explain how to find the "components," i.e., the natural segments of a given sentence. As mentioned above, the identification of appropriate segments within a sentence depends on the task at hand. Nonetheless, there are two main tools that have been used by previous research, namely Language Models and Pointwise Mutual Information. We also use these tools in the search for the natural segmentation of a given sentence. The first research question that is addressed in this chapter is the following:

RQ3 *Given a sentence in some language, identify what conditions a segmentation of the sentence should satisfy, in order for linear compositionality of meaning to hold. How can one define the segmentation that satisfies those conditions optimally?*

Let S be a sentence consisting of n words in any language. As in Chapter 3, a segmentation σ of S is a consecutive partition of S. Our goal is to find the most appropriate segmentation of S; appropriateness is determined by some function F_S with domain the set of S's segmentations and range the real numbers. In other words we want to find

$$\sigma^* = \arg\max_{\sigma} \; F_S(\sigma), \tag{4.1}$$

where the search is performed over all 2^{n-1} possible segmentations. We primarily show what F_S should be such that (4.1) produces the most natural segmentation. Secondarily, it is explained how σ^* can be found efficiently, regardless of the details of F_S. Before we outline the possible choices for F_S, we first explain our notion of a natural segmentation of a sentence.

In order to understand the meaning of a sentence its words should not be inspected in isolation. The combination of the meaning of each word does not, in general, match the meaning of a sentence as a whole. The meaning of a sentence is given by an abstract nonlinear combination of the meaning of its words. Our assertion is that there exists a partition of the words that produces phrases, such that the linear combination of the meaning of these phrases is as close as possible to the meaning of the sentence. From all possible such partitions we consider the largest one to be the one that provides the building blocks of the sentence. For example, for the sentence $S = s_1 ... s_{n-1} s_n$ with $s_n =$"." (the full stop), the two segments of the partition $\{s_1 ... s_{n-1},\ s_n\}$ can provide the meaning of S in an abstract linear combination. But it is very likely that $s_1 ... s_{n-1}$ can be further decomposed into segments that retain linearity of meaning, thus producing a larger partition. Decompositions are performed iteratively until the parts of a partition stop providing the meaning of a sentence linearly.

We do not attempt to formalize the term "linear/nonlinear combination of meaning." It is only used as a means of providing some intuition for our notion of natural building blocks of a sentence. However, if such a formalization is possible, then it should be equivalent to the following more concrete construction: For a given sentence S, let $d(S, S')$ denote a metric of semantic relatedness between S and any other sentence S'. The set

$$D_\epsilon(S) = \{\ S' \ :\ d(S, S') < \epsilon,\ S \neq S'\ \}, \tag{4.2}$$

is thus a collection of sentences whose meaning is different to S up to a tolerance of ϵ. We further assume that words that compose each S' are as different as possible to the ones of S, so that each S' is a true paraphrase of S (up to ϵ), and not just a mildly inflected version of S. As with alignments in SMT, suppose we can align S with each S'. The components of dominant frequency from all such S-to-S' alignments define the building blocks of S. Alternatively, one could consider only

$$S^* = \arg\max_{S' \in D_\epsilon(S)} \; L(S'), \tag{4.3}$$

for some function L that assesses a sentence according to its likelihood. Also in this case the components of the S-to-S^* alignments would define the building blocks of S. This is a more plausible and interesting approach since the abstract notion of meaning is fused with "tangible" statistics from monolingual corpora, i.e., qualitative and quantitative criteria are combined. In both cases the function F_S in (4.1) is implicitly determined by the alignments between S and members of $D_\epsilon(S)$.

The exact formulation of F_S is very difficult as it involves a formulation for $d(S, S')$, the semantic relatedness metric between S and S'. Instead, we try to produce a function that generates the same output as the desired F_S, but without using paraphrases. We follow two approaches.

The first method is based on Language Models (LMs). In n-gram LMs, the order n is fixed, i.e., during training, for any word in a corpus, always its previous $n-1$ words are considered for deriving conditional probabilities. By taking segmentations into account we consider a *varied* n-gram LM: For any word we consider only its history within its segment. For example, for the sentence $S = s_1 ... s_6$, the likelihood of the segmentation $\sigma = (s_1)(s_2 s_3 s_4)(s_5 s_6)$ is

$$l = p(s_1) \times p(s_2) p(s_3|s_2) p(s_4|s_2^3) \times p(s_5) p(s_6|s_5). \tag{4.4}$$

In other words, we apply exactly the intuition of a segment's role in a sentence, namely its ability to act as a distinct component in the sentence. Consequently, we assume varied histories tailored to segments' lengths. Our goal is to find the most appropriate segmentation of S. Another possible segmentation for the same sentence in the above example is $\sigma' = (s_1)(s_2)(s_3 s_4)(s_5)(s_6)$ with likelihood

$$l' = p(s_1) \times p(s_2) \times p(s_3) p(s_4|s_3) \times p(s_5) \times p(s_6). \tag{4.5}$$

We assume that if $l > l'$, then σ is a more appropriate segmentation for S than σ'. It is then easy to show that the most likely segmentation of S is the same as the one that minimizes S's perplexity. This method will be shown not to be in line with the preceding description of natural segments of a sentence. It is, however, of general theoretical interest with potentially useful applications to other areas of SMT, such as ranking of N-best lists of translation candidates during decoding.

The second method that attempts to generate natural segmentations is based on two seemingly unrelated concepts, namely Pointwise Mutual Information (PMI) and partition refinements. The associated research question is as follows:

RQ4 *Given the relationship between Shannon's entropy and metrics on lattices [138], elaborate on the mathematical framework of Pointwise Mutual Information (PMI). Is it possible to extend PMI within this framework for simulating natural segmentations?*

Here, we are interested in assessing the cost of perturbing a segmentation σ into another σ'. A perturbation is assumed to be the smallest possible change that can be applied to σ namely the operation of splitting a single segment of σ into two new segments. Then, for a given sentence, the set of all its possible segmentations equipped with the splitting operation forms a partially ordered set, a lattice in particular. Thus, identifying an appropriate cost function for perturbations is equivalent to identifying a distance metric for partition refinements. We show that the most interesting choice for such a metric is a PMI-like function Δ because it achieves the following goals:

- It gives rise to the desired, most natural segmentation of a sentence. This is achieved by constructing two scoring functions G and F based on Δ; G scores segmentations and F scores segments. These two are related via a simplified version of the Moebius Inversion Formula.

- It provides intuitions behind the successful applications of PMI-based heuristics. It is also known that partition refinements is a valid generalized setting for Shannon's entropy [32, 134]. Thus, our method highlights the previously ignored framework of PMI.

How do we verify that segmentations produced by either varied n-gram LMs or segmentation refinements yield the desired natural segmentations or not? As mentioned above, the construction of (4.2) is difficult. We approximate (4.2) by allowing sentences from different languages to be compared to a given sentence S. In particular, we collect sentences that are translations of S, so that $\epsilon \approx 0$. In practice this is easy to achieve because bitexts are in abundance (at least when compared to, say, English-Paraphrased English parallel corpora). From the proceedings of the European parliament (the so-called Europarl corpus) we found 250K English sentences each of which has translations in 15 other languages. Each such English sentence can be trivially word-aligned to its translation in each of the 15 other languages. In other words, for a sentence S its 15 translations play the role of S's paraphrased sentences in (4.2). The resulting collection of components gives rise to the natural segmentation σ^* of S. We then compare σ^* to the ones generated by our two methods. Overall, we find that partition refinements is the one that produces segmentations that are closer to σ^*.

A matter of secondary importance is the efficient search for σ^* in (4.1), regardless of the details of F_S. Since we restrict ourselves to consecutive partitions of a given sentence with n words, the search space consists of 2^{n-1} possible segmentations. Although this problem could be tackled by using dynamic programming, we employ the Cross-Entropy (CE) method. This is done in order to present a previously unseen link between common problems in natural language processing and importance sampling techniques. Additionally, the CE method can be trivially adapted to the more general problem of extracting bilingual segmentations, which is discussed in Chapter 5.

4.2 Varied n-gram Language Models

As mentioned in Section 2.2, in language modeling a corpus is viewed as a stochastic process: A corpus W is a sequence of words $w_1, .., w_N$; each word w_i is an instance of random variable, or state W_i that takes values from a vocabulary of a certain language. Each state W_i is assumed to be dependent only on its $n-1$ preceding states, or history $W_{i-n+1}, ..., W_{i-1}$. If $n = 2$ then W is a Markov Chain. The quantities of interest are the conditional probabilities

$$p_n(w_i \mid w_{i-m+1}^{i-1}) \;:=\; p_n(W_i = w_i \mid W_{i-m+1} = w_{i-m+1}, ..., \; W_{i-1} = w_{i-1}), \quad (4.6)$$

for all $1 < m \leq n$ and $1 \leq i \leq N$, wherever histories exist. For any word w and any observed sequence of words h, maximizing the log-likelihood of corpus W

$$\begin{aligned}\ell_W(n) &= \log p_n(w_1^N) \\ &= \sum_{i=1}^{N} \log p_n(w_i \mid w_{i-n+1}^{i-1}),\end{aligned} \tag{4.7}$$

results in the estimates

$$p_n(w|h) = \begin{cases} \frac{\text{count}(hw)}{\text{count}(h)}, & \text{if } |h| \leq n-1 \\ \\ \frac{\text{count}(\tilde{h}w)}{\text{count}(\tilde{h})}, & \text{otherwise,} \end{cases} \tag{4.8}$$

where count(s) is the number of times that sequence s has been observed in the corpus, and $\tilde{h} \subset h$ consists of h's final $n-1$ words. See Appendix A for a detailed derivation.

The corpus W is assumed to be universal, i.e., it consists of various topics and/or genres of a language. The optimal choice for n, say n^*, is done with the aid of a test corpus. Such a corpus, say C, supposedly represents the domain of interest; it can be either a subset of W or a different corpus. If C consists of words $v_1, ..., v_M$, then n^* is typically computed as

$$n^* = \underset{n}{\text{argmin}} \quad 2^{-\frac{1}{M}\sum_{i=1}^{M} \log p_n(v_i \mid v_{i-n+1}^{i-1})}, \tag{4.9}$$

where the probabilities are given by (4.8) and the search space is the positive natural numbers. The quantity $2^{-\frac{1}{M}\ell_C(n)}$ in (4.9) is known as perplexity in computational linguistics and its origins lie in the Shannon-McMillan-Breiman Theorem [32]. It states that, for almost all corpora with M words that satisfy some mild conditions, the quantity $-\frac{1}{M}\log p(v_1^M)$ tends to the expected number of bits per word that are required to describe the corpus, as $M \to \infty$. The exponentiation of $-\frac{1}{M}\ell_C(n)$ is loosely associated with the so-called typical set [32], but is purely cosmetic. In fact, the argument that maximizes the likelihood $\ell_C(n)$ yields the same n^*. Once n^* is found, one would apply this n^*-gram LM on various corpora $C', \; C'', ...$ for whatever is the task in hand (e.g., topic modeling). Since the number of words $M', \; M'', ...$ may differ, then average log-likelihood and, traditionally, perplexity is the standard measure for comparisons.

If $C \not\subseteq W$ then it is possible that a sequence of words h may not have been observed in W, i.e., $h \in C$ but $h \notin W$. In this case $p_n(w|h)$ is undefined for any w, for certain n. As mentioned in Chapter 2, such problems are overcome with smoothing, i.e., by employing techniques that assign probability mass for unseen sequences.

In SMT the typical choice for n is between three and five. Rather than an outcome of (4.9), it is a range of values that has empirically been found to allow good synchronization of the LM with the other models of SMT. In this section we challenge the notion of fixed memory in LMs. Although (4.9) is also irrelevant here, perplexity still plays a central role. We proceed with defining varied n-gram LMs.

Suppose we allow the memory to be as large as possible, so that the maximum likelihood estimates of (4.8) become

$$p(w|h) = \frac{\text{count}(hw)}{\text{count}(h)}, \tag{4.10}$$

for any word w and any sequence of words h that has been observed in corpus W. Let C be another corpus. Using (4.10), for any word w in C, our goal is to find how much memory is sufficient for w, or equivalently how much memory is indifferent when predicting w.

We elaborate on our point with an example. Suppose that corpus C has a sequence of words $a...g$. If a standard trigram LM was used, then C's perplexity would be calculated based on the subsequences of Figure 4.1(a). A rectangle indicates how much memory should be used for predicting the rightmost word in that rectangle, so that $p_3(d|...abc) = p_3(d|bc)$, etc. By considering varied memory, *one possible configuration* of rectangles, could be as in Figure 4.1(b). Observe the lack of memory transitivity within fragment $defg$: $g \to f \to e$ and $g \to d$, but $f \not\to d$ and $e \not\to d$, where an arrow is read as "depends on." It provides the interesting interpretation of discontinuous, or

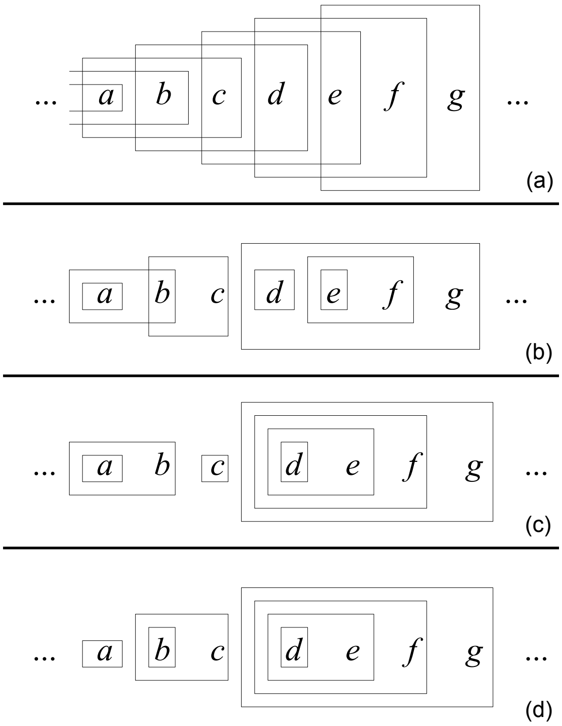


Figure 4.1: (a) Standard trigram LM. (b) General varied n-gram LM. (c) Varied n-gram LM with memory transitivity. (d) Similar to (c), but for a different configuration.

skipping, LMs [58, 69, 107, 127]. The joint probability for $defg$ is given by

$$p(defg) = p(d)p(e)p(f|e)p(g|dX_1X_2), \tag{4.11}$$

where X_1 and X_2 are wild card tokens, so that $p(g|dX_1X_2)$ is read as "probability of g given d followed by any two other words". In order to compute probabilities like $p(g|dX_1X_2)$ one has to collect counts of discontinuous sequences from W. Since

- we do not impose any restrictions on varied memories in a configuration of C and
- typically W consists of tens of millions of sentences,

the collection of such counts from W is impossible in practice. We thus disallow varied n-gram configurations of C that include discontinuous memories. We simplify even further and disallow configurations that do not respect transitivity of memory in general, and not just within a memory rectangle. For instance, the fragment abc in Figure 4.1(b) satisfies $c \to b \to a$ but $c \not\to a$. Two possible simplified configurations of Figure 4.1(b) are shown in Figure 4.1(c) and (d). Hence the problem of finding the most appropriate varied n-gram configuration of a corpus C is reduced to finding the most appropriate segmentation for C. Once such a segmentation is found, recovering the nested memory dependencies within each segment is trivial.

In general, if corpus C consists of the sequence of words $v_1, ..., v_M$, then a segmentation ω of C is a relabeling of C's words as in

$$\begin{array}{c} v_1 \quad v_2 \quad \cdots \quad v_{M-1} \quad v_M \\ \searrow \\ \omega_{11}\,\omega_{12} \ldots \omega_{1M_1} \quad \omega_{21}\,\omega_{22} \ldots \omega_{2M_2} \quad \cdots \quad \omega_{k1}\,\omega_{k2} \ldots \omega_{kM_k} \end{array}$$

so that C is partitioned into k parts (segments). Each part i consists of M_i words that are consecutive in C. Clearly, we have $M = \sum_{i=1}^{k} M_i$. As explained in the example above, each segment indicates how much memory is sufficient in order to predict a word. Our goal is to find the best possible segmentation of C using the maximum likelihood estimates (4.10). The log-likelihood of a segmentation ω of C with k^ω segments is given by

$$\ell_C(\omega) = \sum_{i=1}^{k^\omega} \sum_{j=1}^{M_i^\omega} \log p\left(\omega_{ij} |\, \omega_{i1}^{i(j-1)}\right), \tag{4.12}$$

where the probabilities are given by (4.10) and M_i^ω denotes the number of words in part i under segmentation ω. We assume that the optimal segmentation ω^* achieves the highest value for $\ell_C(\omega)$, i.e.,

$$\omega^* = \underset{\omega}{\text{argmin}} \;\; 2^{-\frac{1}{M}\ell_C(\omega)}, \tag{4.13}$$

where the search is over all 2^{M-1} segmentations of C.

The above optimization problem can be simplified by considering each sentence of the corpus individually; finding the optimal segmentation of a sentence is independent of all other sentences. The optimal segmentation σ^* of a single sentence S with n words is given by

$$\sigma^* = \arg\max_{\sigma} \ \ell_S(\sigma), \tag{4.14}$$

where $\ell_S(\sigma)$ is calculated in the same way as in (4.12) and the search is over all 2^{n-1} possible segmentations of S. If C consists of L sentences $S_1 ... S_L$, then the optimal segmentation of the whole corpus is

$$\omega^* = (\sigma_1^*, ..., \sigma_L^*), \tag{4.15}$$

where σ_l^* is the optimal segmentation of sentence S_l. Finally, the perplexity of the optimally segmented corpus is $2^{-\frac{1}{M}\ell_C(\omega^*)}$, where

$$\ell_C(\omega^*) = \sum_{l=1}^{L} \ell_{S_l}(\sigma_l^*). \tag{4.16}$$

In practice, we consider only 6 words as the maximum possible memory for a given word. This implies that for the computation of (4.10) we have to collect counts form all n-grams with $n = 1, ..., 7$.

In contrast with standard n-gram LMs, smoothing is not necessary for varied n-gram LMs. If the training corpus W is large, then an unobserved sequence hw in C, provides strong indication that hw should not appear as a segment in C. For the special case where $w \in C$ but $w \notin W$, then the word w will appear as segment on its own in any possible segmentation ω of C; all other configurations that include w in larger segments are disallowed.

From the formulation of varied n-gram LMs it is not possible to determine the type of segments that are formed in C. We leave that for Section 4.5.

4.3 PMI and segmentation refinements

In this section the connection between PMI and metrics on partial orders is established. The set of all possible segmentations of a sentence together with the operation of "segmentation refinement" is known to form a particular type of partial order, namely a lattice. We provide the set up for the well-known formulation of metrics on lattices based on valuations, i.e., functions that score elements of lattices. A certain choice for valuations yields the desired link. Finally, equipped with such a metric we explain how to simultaneously score segments and segmentations of a sentence.

4.3.1 Metrics on lattices

A partially ordered set, or poset, is the mathematical setup that allows the examination of order in countable sets. Formally a poset is a pair $(P, \leq)$, where P is a countable set and "$\leq$" is a binary relation that satisfies for all $x, y, z \in P$:

- $x \leq x$ (reflexivity);
- if $x \leq y$ and $y \leq x$ then $x = y$ (antisymmetry);
- if $x \leq y$ and $y \leq z$ then $x \leq z$ (transitivity).

A poset may contain pairs of elements $x, y \in P$ that are incomparable, i.e., neither $x \leq y$ nor $y \leq x$ holds. The elements of a poset can be anything (numbers, sets,...) and the binary relation "$\leq$" is interpreted according to the poset in consideration. Nonetheless, the intuition of "greater than" or "dominance" is carried over in all cases.

Every finite poset has a visual representation that is known as Hasse diagram. Figure 4.2 shows examples of such diagrams. The arrows on edges are used to emphasize how two elements of the poset are related. For instance, in the leftmost poset of Figure 4.2 we have $x_6 \leq x_8$ and similarly for all other arrows.

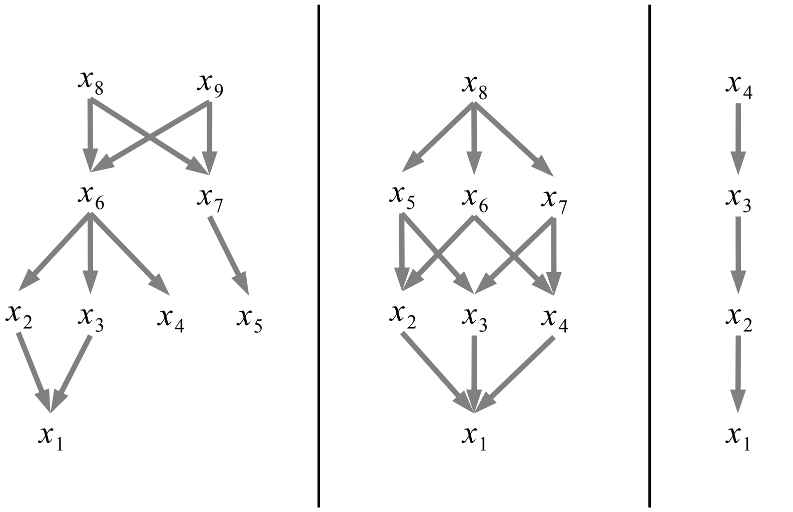


Figure 4.2: Examples of posets visualized by Hasse diagrams.

An element $u \in P$ is an upper bound of a subset $Q \subseteq P$ if for all $x \in Q$, then $x \leq u$. An element $l \in P$ is a lower bound of a subset $Q \subseteq P$ if for all $x \in Q$, then $l \leq x$. Due to transitivity there may be multiple upper bounds and/or lower bounds for some $Q \subseteq P$; see Figure 4.3 for an example.

Let U_Q (L_Q) denote the set of all upper (lower) bounds of Q. The supremum of Q, denoted by $\sup Q$, is the upper bound of Q such that $\sup Q \leq u$, for all $u \in U_Q$. Similarly, the infimum of Q, denoted by $\inf Q$, is the lower bound of Q such that

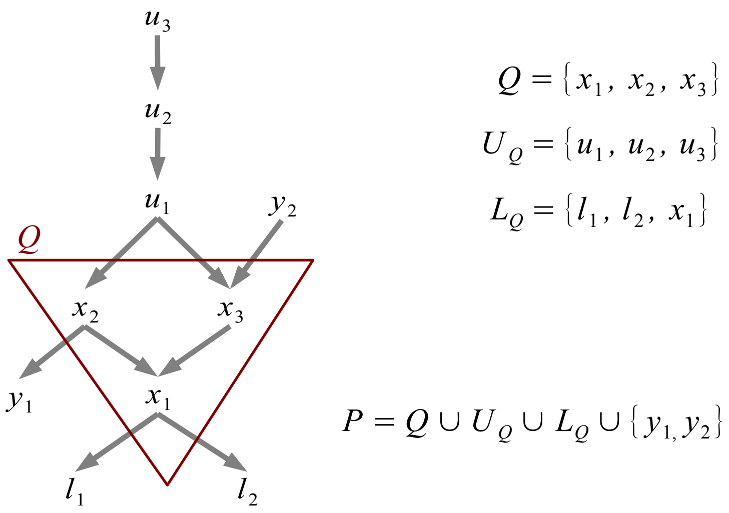

u_3
u_2
u_1
y_2
Q
x_2
x_3
y_1
x_1
l_1
l_2
$Q = \{x_1, x_2, x_3\}$
$U_Q = \{u_1, u_2, u_3\}$
$L_Q = \{l_1, l_2, x_1\}$
$P = Q \cup U_Q \cup L_Q \cup \{y_1, y_2\}$

Figure 4.3: All upper and lower bounds of $Q \subset P$, denoted by U_Q and L_Q respectively. Observe that x_1 is both an element of Q and a lower bound for Q.

$l \leq \inf Q$, for all $l \in L_Q$. Note that $\sup Q$ and/or $\inf Q$ may not exist; see Figure 4.4 for a counterexample.

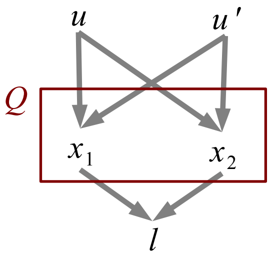


Figure 4.4: An example of $Q \subseteq P$ for which $\sup Q$ does not exist.

A lattice is a poset P such that for any two elements $x, y \in P$, both $\sup\{x, y\}$ and $\inf\{x, y\}$ exist and we define

$$x \vee y = \sup\{x, y\}, \quad \text{the join of } x \text{ and } y;$$
$$x \wedge y = \inf\{x, y\}, \quad \text{the meet of } x \text{ and } y.$$

Lattices play central role in order theory. We focus on partition refinements [142], a lattice that is formed by the set of partitions of a set S equipped with a inclusion-based binary relation, which is defined as follows: A partition σ' is said to be finer than a partition σ, if every part of σ' is a subset of a part of σ, and we write $\sigma' \leq \sigma$. For our purposes S is a sentence consisting of words and a partition of S is a grouping of S's words into non-overlapping, possibly discontinuous, phrases. For practical reasons that will be explained in Section 4.5, we focus on a subposet of partition refinements, namely segmentation refinements [63]. In other words, we disallow partitions that include discontinuous phrases. Figure 4.5 shows a poset of segmentation refinements for a sentence with four words. In general, although a subposet of a lattice is not necessarily a lattice, it is in this case easy to show that segmentation refinements is also a lattice, but we omit the details.[1]

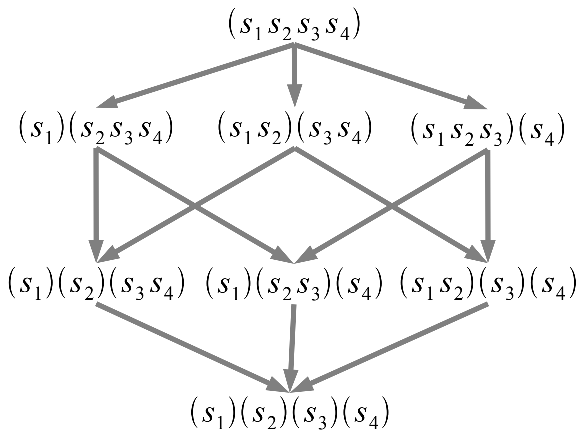


Figure 4.5: Hasse diagram of segmentation refinements for a sentence with four words. Brackets are used to distinguish different segments.

Finally, metrics on lattices are formulated via valuations [105]. A valuation on a lattice P is a function $v : P \to \mathbb{R}$ that satisfies

$$v(x) + v(y) = v(x \vee y) + v(x \wedge y), \tag{4.17}$$

for all $x, y \in P$. A valuation is called isotone if, for all $x, y \in P$ with $x \leq y$, we have $v(x) \leq v(y)$. It is called antitone if, for all $x, y \in P$ with $x \leq y$, we have $v(x) \geq v(y)$. For any $x, y \in P$ the distance function

$$d(x,y) = \begin{cases} v(x \vee y) - v(x \wedge y), & \text{if } v \text{ is isotone} \\ v(x \wedge y) - v(x \vee y), & \text{if } v \text{ is antitone,} \end{cases} \tag{4.18}$$

is a metric [12].

4.3.2 Segmentation log-likelihood as a valuation

There is another more intuitive construction of segmentation refinements. By fixing a segmentation σ, we are interested in perturbing σ and generating another segmentation

[1] The poset of segmentation refinements of a set S is isomorphic to ordering the powerset of S by inclusion. The latter is a well-known lattice [142].

σ'. Such a perturbation is achieved by splitting a single segment of σ into two new segments, while keeping all other segments fixed. For example, for a sentence with five words, if $\sigma : (s_1 s_2)(s_3 s_4 s_5)$, where brackets are used to distinguish the segments $s_1 s_2$ and $s_3 s_4 s_5$, then σ can be perturbed in three different ways:

- $\sigma' : (s_1)(s_2)(s_3 s_4 s_5)$, by splitting the first segment of σ.
- $\sigma'' : (s_1 s_2)(s_3)(s_4 s_5)$, by splitting at the first position of the second segment of σ.
- $\sigma''' : (s_1 s_2)(s_3 s_4)(s_5)$, by splitting at the second position of the second segment of σ,

so that σ', σ'' and σ''' are the perturbations of σ. Clearly, these resulting segmentations can also be perturbed in the same way. It is trivial to show that the set of all segmentations of a sentence equipped with this splitting operation forms indeed the poset of segmentation refinements.

We focus on pairs of segmentations (σ, σ') such that σ' is a perturbation of σ. In this case we always have

$$\sigma \vee \sigma' = \sigma \quad \text{and} \quad \sigma \wedge \sigma' = \sigma', \tag{4.19}$$

so that the isotone case of (4.18) is rewritten as

$$d(\sigma, \sigma') = v(\sigma) - v(\sigma'), \tag{4.20}$$

and similarly if v is antitone. (4.20) is interpreted as the cost of perturbing σ in order to generate σ'.

Before we present some choices for valuations we first fix some notation for segmentations. A segmentation σ is treated as a multiset with each member of σ being a segment. If F is any function whose domain is phrases/segments then $\sum_{i=1}^{|\sigma|} F(x_i) = \sum_{x \in \sigma} F(x)$, where $|\sigma|$ is the number of segments in σ. The multiplicity of each segment x is implicitly taken into account, because σ is assumed a multiset. If σ' is a perturbation of σ, then there exists a pair of segments in σ' that resulted from a single segment in σ. We write $(s, \bar{s})$ for such a pair, so that $s\bar{s}$ is the unperturbed segment.

An interesting choice for $v(\sigma)$ in (4.20) is

$$v(\sigma) = \sum_{x \in \sigma} \log p(x), \tag{4.21}$$

with

$$p(x) = \frac{\text{count}(x)}{\sum\limits_{x' \in W} \text{count}(x')}, \tag{4.22}$$

where W is a corpus and count(x) is the frequency of x in W (only up to 7-grams inclusive are considered in W). Without worrying whether (4.21) is isotone or antitone, if σ' is a perturbation of σ, then (4.20) becomes

$$\begin{aligned} d(\sigma,\sigma') &= \sum_{x\in\sigma}\log p(x) - \sum_{x\in\sigma'}\log p(x) \\ &= \left(\log p(s\bar{s}) + \sum_{x\in\sigma\setminus\{s\bar{s}\}}\log p(x)\right) - \left(\log p(s) + \log p(\bar{s}) + \sum_{x\in\sigma'\setminus\{s,\bar{s}\}}\log p(x)\right), \end{aligned} \tag{4.23}$$

where $\tilde{\sigma} \equiv \sigma \setminus \{s\bar{s}\} = \sigma' \setminus \{s,\bar{s}\}$, because the unperturbed segments of σ remain intact in σ'. Thus, (4.23) becomes simply

$$d(\sigma,\sigma') = \log\frac{p(s\bar{s})}{p(s)p(\bar{s})}, \tag{4.24}$$

which is the usual Pointwise Mutual Information. However, (4.24) is problematic because it is independent of $\tilde{\sigma}$. Since there exist multiple pairs (σ,σ') that perturb the same segment $s\bar{s}$ into s and $\bar{s}$, all such pairs will have the same cost of perturbing σ into σ'. This is unwanted as we wish to establish metrics between segmentations and not just segments. This could be tackled by considering average log-likelihood as a valuation, i.e.,

$$v(\sigma) = \frac{1}{|\sigma|}\sum_{x\in\sigma}\log p(x), \tag{4.25}$$

which results in

$$d(\sigma,\sigma') = \log\frac{p(s\bar{s})^{\frac{1}{|\sigma|}}}{(p(s)p(\bar{s}))^{\frac{1}{|\sigma|+1}}} \quad + \quad \frac{1}{|\sigma|(|\sigma|+1)}\sum_{x\in\tilde{\sigma}}\log p(x), \tag{4.26}$$

where we used the fact that $|\sigma'| = |\sigma| + 1$. In practice both (4.24) and (4.26) were found to be inadequate within the framework that will be described in the rest of this chapter.

The valuation that was found to be useful in practice is given by

$$v(\sigma) = \sum_{x\in\sigma}\log p_\sigma(x), \tag{4.27}$$

with

$$p_\sigma(x) = \frac{\text{count}(x)}{\sum\limits_{x'\in\sigma}\text{count}(x')}, \tag{4.28}$$

where count(x) is the frequency of x in some corpus W. By letting $N_\sigma = \sum_{x\in\sigma}\text{count}(x)$, we find

$$d(\sigma,\sigma') = \log\frac{p_\sigma(s\bar{s})}{p_{\sigma'}(s)p_{\sigma'}(\bar{s})} \quad + \quad (|\sigma|-1)\log\frac{N_{\sigma'}}{N_\sigma}. \tag{4.29}$$

In particular, the function that assesses the cost of perturbing $s\bar{s}$ in σ and generating s and $\bar{s}$ in σ', as defined by

$$\Delta(\sigma \rightarrow \sigma') = \log \frac{p_{\sigma}(s\bar{s})}{p_{\sigma'}(s)p_{\sigma'}(\bar{s})}, \tag{4.30}$$

will be shown to be useful for identifying the most natural segmentation of a sentence. We henceforth write $\sigma \rightarrow \sigma'$ if σ' is a perturbation of σ. We also write $\Delta_{s\bar{s}}(\sigma \rightarrow \sigma')$ for the left hand side of (4.30) whenever we want to emphasize the split for which the perturbation $\sigma \rightarrow \sigma'$ is responsible.

4.3.3 Relationship between segments and segmentations

As mentioned in the previous section, the splitting of a segment $s\bar{s}$ into two new segments s and $\bar{s}$ can be a result of different perturbations $\sigma \rightarrow \sigma'$. For example, for the sentence $S = x_1^3 s x_4^8$, where s is a sequence of words, we have

$$\begin{array}{rl} \sigma_1 \; : & (x_1x_2x_3)(\mathbf{s}\;\mathbf{x_4x_5})(x_6x_7)(x_8) \\ & \qquad\qquad\qquad \searrow \\ \sigma'_1 \; : & (x_1x_2x_3)(\mathbf{s})(\mathbf{x_4x_5})(x_6x_7)(x_8) \end{array}$$

and

$$\begin{array}{rl} \sigma_2 \; : & (x_1)(x_2x_3)(\mathbf{s}\;\mathbf{x_4x_5})(x_6)(x_7x_8) \\ & \qquad\qquad\qquad \searrow \\ \sigma'_2 \; : & (x_1)(x_2x_3)(\mathbf{s})(\mathbf{x_4x_5})(x_6)(x_7x_8) \end{array}$$

among all possible perturbations. In both cases the operation of splitting sx_4x_5 into s and $\bar{s} = x_4x_5$ is responsible for perturbations $\sigma_1 \rightarrow \sigma'_1$ and $\sigma_2 \rightarrow \sigma'_2$. The costs $\Delta(\sigma_1 \rightarrow \sigma'_1)$ and $\Delta(\sigma_2 \rightarrow \sigma'_2)$ will not in general be the same and depend on $\tilde{\sigma_1}$ and $\tilde{\sigma_2}$. By inspecting all such pairs $\sigma \rightarrow \sigma'$ for which the same s and $\bar{s}$ are 'co-responsible' we could learn whether s and $\bar{s}$ should appear together as a segment in S. In fact, we inspect all cases in which segment s appears as a conjugate in a co-responsible pair. In the example above, we could also have

$$\begin{array}{rl} \sigma_3 \; : & (x_1)(\mathbf{x_2x_3s})(x_4x_5x_6x_7x_8) \\ & \qquad\qquad \searrow \\ \sigma'_3 \; : & (x_1)(\mathbf{x_2x_3})(\mathbf{s})(x_4x_5x_6x_7x_8) \end{array}$$

in which case s and $\bar{s} = x_2x_3$ are co-responsible for $\sigma_3 \rightarrow \sigma'_3$. Since similar inspections can be done for any segment, we can then decide which are the best segments and finally identify the optimal segmentation σ^* of S.

In general, the cost function $\Delta_{s\bar{s}}(\sigma \to \sigma')$ admits a measure for the segments that are co-responsible for perturbing σ into σ'; we define the gain of s from the perturbation $\sigma \to \sigma'$ as

$$gain_{\sigma\to\sigma'}(s) = -\Delta_{s\bar{s}}(\sigma \to \sigma'). \tag{4.31}$$

Segment s may be co-responsible for different perturbations, and we consider all such perturbations. Let

$$R(s) = \{\sigma \to \sigma' : \ s \notin \sigma, \ s \in \sigma'\} \tag{4.32}$$

denote the set of perturbations for which s is a conjugate in a co-responsible pair. Then, the average gain of s in the sentence is given by

$$gain(s) = \frac{1}{|R(s)|} \sum_{\{\sigma\to\sigma'\}\in R(s)} gain_{\sigma\to\sigma'}(s). \tag{4.33}$$

Intuitively, $gain(s)$ measures how difficult it is to break phrase s into sub-phrases. Finally, the quality measure of a segmentation σ of a sentence is given by

$$g(\sigma) = \sum_{s\in\sigma} gain(s), \tag{4.34}$$

and refer to $g(\sigma)$ as the surface measure of segmentation σ for the given sentence. Note that g is a real number. The relation $g(\sigma) > g(\rho)$ implies that σ is a better segmentation than ρ.

The exact computation of $gain(s)$ for each possible segment s is computationally expensive since all possible perturbations need to be considered. In preliminary experiments we found the following approximation to work well:

1. Generate a random sample of no more than 1500 segmentations.

2. Perturb each such segmentation in all possible ways.

3. Consider each segment s that is co-responsible in those perturbations and compute $gain(s)$ based on those perturbations only.

Moreover, only segments of certain length are of interest in practice: Segments that consist of up to 7 words are considered in Step 3; the rest are ignored.

The choices for (4.30)–(4.34) are heuristics but the experiments in the following sections, together with the existing setup, provide evidence for further formalization. The segment-segmentation relationship of this section can be viewed as an approximated version of the Moebius Inversion Theorem. It states that, for any poset P, there exist functions $F : P \to \mathbb{R}$ and $G : P \to \mathbb{R}$, such that the following statements are equivalent:

1. $G(y) = \sum_{x\leq y} F(x), \quad$ for all $y \in P$;

2. $F(y) = \sum_{x \leq y} \mu(x,y)\, G(x), \quad$ for all $y \in P$,

for some function $\mu : P \times P \to \mathbb{R}$ with the property $\mu(x,y) = 0$, if x, y are incomparable [142]. We leave such investigations for future work. Instead, given sentence S, we focus on finding the optimal segmentation

$$\sigma^* = \arg\max_{\sigma} g_S(\sigma), \tag{4.35}$$

where g_S is based on (4.30)–(4.34).

4.4 The Cross-Entropy method for optimal segmentations

Given a sentence that consists of n words and a performance function F that scores segmentations of the sentence, we show how to compute

$$\sigma^* = \arg\max_{\sigma} F(\sigma)$$

efficiently. To this end the Cross-Entropy (CE) method for combinatorial optimization is employed [128]. In Appendix B we provide the importance sampling techniques that motivate the algorithm of the CE method, as well as the algorithm itself. We proceed with explaining how to find the optimal segmentation σ^* as determined by a scoring function F for a given sentence.

In order to apply the CE method for our purposes, we first need to find the appropriate pmf from which random segmentations are drawn. This task can be carried out easily if the following observation is taken into account. A segmentation of a given sentence has a bit string representation in the following way: If two consecutive words in the sentence belong to the same segment in the segmentation, then this pair of words is encoded by '1', otherwise by '0'. Figure 4.6 shows the bit string representations of all segmentations of a sentence with four words.

In general, it is trivial to show that the set of all segmentations of a sentence with n words is in one-to-one correspondence with the set of all bit strings of length $n-1$. The bijection, say B, is given by the map described above.

Given a sentence with n words, the problem of finding the optimal segmentation, with respect to a scoring function F, is equivalent to finding the optimal bit string of length $l = n - 1$, with respect to $F' = F \circ B^{-1}$. For simplicity we write F for both scoring functions, and, for the rest of this section, we do not distinguish between a segmentation and its bit string representation.

Let X denote the set of all bit strings of length l. A pmf $f(\cdot;\theta)$ of X parametrized by θ is defined as follows. A random bit string x is a sequence $x_1 ... x_l$ of l independent

$(w_1 w_2 w_3 w_4)$		111
$(w_1 w_2 w_3)(w_4)$		110
$(w_1 w_2)(w_3 w_4)$		101
$(w_1)(w_2 w_3 w_4)$	$\Leftrightarrow$	011
$(w_1 w_2)(w_3)(w_4)$		100
$(w_1)(w_2 w_3)(w_4)$		010
$(w_1)(w_2)(w_3 w_4)$		001
$(w_1)(w_2)(w_3)(w_4)$		000

Figure 4.6: All segmentations of sentence w_1^4 (left) and all bit strings of length three (right). Each row shows the correspondence between a segmentation and a bit string.

Bernoulli random variables, i.e.,

$$x_j = \begin{cases} 1, & \text{with probability } \theta_j \\ \\ 0, & \text{with probability } 1 - \theta_j, \end{cases} \tag{4.36}$$

for all $j = 1, ..., l$. Parameter θ is thus a vector of probabilities $(\theta_1, ..., \theta_l)$. Since the value of a bit is assumed independent of all other bits in a bit string, the pmf is given by

$$f(x;\theta) = \prod_{j=1}^{l} \theta_j^{x_j} (1-\theta_j)^{1-x_j}. \tag{4.37}$$

If no further information is known about X, i.e., if $\theta = (1/2, ..., 1/2)$, then $f(x;\theta) = 1/2^l = 1/|X|$, for all $x \in X$. This instance for θ will be used as the initial value θ^0 in Step 1 of the algorithm.

In order to update θ at each iteration of the algorithm we need to solve the maximization problem (B.13) in Step 4. In other words, given $\gamma \in \mathbb{R}$ and positive integer N, we want to find the argument that maximizes

$$J(\theta) = \frac{1}{N} \sum_{i=1}^{N} I_{\{F(x_i) \geq \gamma\}} \ \ln f(x_i;\theta), \tag{4.38}$$

where I and f are given by (B.3) and (4.37) respectively, and $\{x_1, ..., x_N\}$ is a sample of bit strings. We denote by x_{ij} the jth bit of the ith instance. It is easy to show, see for example [34], that this optimal vector of probabilities is given by

$$\theta_j = \frac{\sum_{i=1}^{N} I_{\{F(x_i)\geq\gamma\}} \;\; x_{ij}}{\sum_{i=1}^{N} I_{\{F(x_i)\geq\gamma\}}}, \qquad \text{for all } j = 1, ..., l. \tag{4.39}$$

In words, θ_j equals

$$\frac{\text{\# instances performing at least } \gamma \text{ and have `1' at } j\text{th bit}}{\text{\# instances performing at least } \gamma}, \tag{4.40}$$

because $x_{ij} \in \{0, 1\}$ for all $i = 1, ..., N$ and $j = 1, ..., l$. Fix N such that $N/|X| \ll 1$ and set $\rho = \lceil 1\% N \rceil$. Also, let $\alpha \in (0, 1)$ denote the smoothing parameter. The algorithm for finding the best performing bit string under a scoring function F is as follows.

1. Set $(\theta_1^0, ..., \theta_l^0) = (1/2, ..., 1/2)$.

2. Generate sample $x_1, ..., x_N$ of bit strings, each of length l, such that $x_{ij} \sim$ Bernoulli(θ_j^{t-1}), for all $i = 1, ..., N$ and $j = 1, ..., l$. Compute scores $F(x_1), ..., F(x_N)$. Order them descendingly as $F(x_{\pi(1)}) \geq ... \geq F(x_{\pi(N)})$, where π is the associated permutation of $\{1, ..., N\}$.

3. Compute performance threshold $\gamma^t = F(x_{\pi(\rho)})$.

4. Compute bit probabilities

$$\theta_j^t = \frac{\sum_{i=1}^{N} I_{\{F(x_i)\geq\gamma^t\}} \;\; x_{ij}}{\sum_{i=1}^{N} I_{\{F(x_i)\geq\gamma^t\}}}, \qquad j = 1, ..., l. \tag{4.41}$$

5. Smooth bit probabilities

$$\theta_j^t := \alpha\theta_j^t + (1-\alpha)\theta_j^{t-1}, \qquad j = 1, ..., l. \tag{4.42}$$

6. If $t \geq 5$ and $\gamma^t = \gamma^{t-1} = ... = \gamma^{t-5}$, then stop. Else $t := t + 1$ and go to Step 2.

Why is smoothing necessary? At each iteration t, the cut-off parameter ρ forces the selection of a small subsample that eventually yields performance threshold γ^t. Regardless of what γ^t is, ρ is small enough to prematurely yield bit probabilities $\theta_j^t = 0$ or $\theta_j^t = 1$ for certain j in (4.42). More precisely, in practice a value of $\rho = 5$ is typical for bit strings of length ~ 20, i.e., when the sentence consists of ~ 20 words. Even in the initial iterations of the algorithm, say for some t, it is very likely that all 5 best-performing instances of sample O^t may have a '0' at the jth bit, for some j. This results in $\theta_j^t = 0$ and, consequently, no instance of $O^{t+\tau}$ will have a '1' at position j, for all $\tau \geq 1$. Similar problem happens when $\theta_j^t = 1$ at an early iteration t, for some j. Such premature convergence may lead to poor local optima. Thus at every iteration the new and old bit probabilities are interpolated with smoothing parameter α. Surely, more mass should be assigned to new probabilities; in practice we have found $\alpha = 0.7$ to perform well, which is in line with the suggested value in the literature [129].

At $t = t_\infty$, vector θ^{t_∞} consists entirely of '0's and '1's. Ideally, $\gamma^{t_\infty} = \gamma^*$ and $\theta^{t_\infty} = x^*$. By setting $N = 20l$ for the sample size we found ideal or nearly ideal scenarios using either ℓ_S of (4.14) or g_S of (4.35) as performance functions for a given sentence S.

4.5 The natural segmentation of a sentence

In this section the natural segmentation of a sentence is defined. In order to compute it in practice, one needs to generate paraphrases for each segment of a segmentation of a given sentence. As this is a difficult task we turn to an approximation by focusing on aligned bitexts. We explain that a translation T of a sentence S together with alignments between S and T can be used as an approximation for the natural segmentation of S. To this end, using the Europarl corpus, the natural segmentation of approximately 250K English sentences are approximated. For these sentences optimal segmentations as in (4.14) or (4.35) are also computed; basic comparisons provide strong indications that optimal poset-based segmentations are closer in simulating natural segmentations.

4.5.1 Definition of natural segmentation

Let S be sentence in a language, i.e., a sequence of words that respects the syntax of a language. Let $\sigma = \{x_1, ..., x_m\}$ denote a segmentation of S. That is, σ is such that

- Each segment $x \in \sigma$ is a sequence of words which respects the order of appearence in S, but may be discontinuous.
- There exists an ordering of all segments that produces sentence S, or $O_\sigma = S$ for short. Such an ordering also merges discontinuous segments with other segments appropriately.

For any segment x let $Para(x)$ denote the set of x's paraphrases. For each segment x_i of σ we can choose a paraphrase $y_{ij} \in Para(x_i)$ and form the 'paraphrased segmentation' $\rho = \{y_{1j_1}, ..., y_{mj_m}\}$. Clearly there are $\prod_{i=1}^{m} |Para(x_i)|$ ways of constructing such configurations of paraphrased segmentations.

Consider the subset of all possible configurations

$$P = \{\rho \mid \exists \text{ ordering } O_\rho \text{ that produces a sentence}\}. \tag{4.43}$$

We wish to compare S with the sentences that can be generated by P. As mentioned in Section 4.1, the measure of comparison is some semantic relatedness metric $d(S, S')$, for any other sentence S'. The quantity of interest is

$$\delta = \max_{\rho \in P} d(S, O_\rho), \tag{4.44}$$

i.e., the largest possible dissimilarity (worst case scenario) that can be achieved. We write P_σ and δ_σ for P and δ respectively whenever emphasis is needed.

If σ contains small segments then δ is more likely to be large: The smaller segment x is, the more likely that members of $Para(x)$ will produce ambiguity in a paraphrased sentence of S. On the other hand, if σ contains larger segments then δ is more likely to be small, for similar reasons. This bias towards larger segments can be removed by imposing restrictions on the quality of segments: Given an empirical distribution p, for any segment x, if $p(x) > \theta$, for some threshold θ, then x is allowed to be part of a segmentation. As a shorthand we write $p(\sigma) > \theta$ for $p(x_i) > \theta$, $i = 1, ..., |\sigma|$.

The natural segmentation of a sentence S in some language is defined as

$$\begin{aligned} \sigma^* &= \operatorname*{argmin}_{\sigma \,:\, p(\sigma)>\theta} \max_{\rho \in P_\sigma} d(S, O_\rho) \\ &= \operatorname*{argmin}_{\sigma \,:\, p(\sigma)>\theta} \delta_\sigma. \end{aligned} \tag{4.45}$$

Without the restriction $p(\sigma) > \theta$, the search space is over all continuous and discontinuous segmentations, i.e., partitions of S. If S consists of n words then there are B_n such partitions, where B_n is the nth Bell number and is computed as

$$B_n = \sum_{k=0}^{n} \left\{ n \atop k \right\}, \tag{4.46}$$

where $\left\{ n \atop k \right\}$ are Stirling numbers of the second kind [31]. Even with the restriction $p(\sigma) > \theta$ obvious complications may arise when solving (4.45) in practice. For simplicity we allow continuous segmentations only, whose full search space consists of 2^{n-1} possible configurations. If a sentence consists of twenty words, then there are approximately half a million such configurations. On the other hand, we have[2] $B_{20} > 50 \cdot 10^{12}$, which justifies the need for this simplification.

The definition of natural segmentation is motivated by the construction of measure theoretic entropy in dynamical systems [77, 120]. A discussion on the possible connections between natural sentence segmentations and such an entropy (as well as the more general Kolmogorov-Sinai entropy, which does not invoke a metric) is beyond the scope of this thesis. We simply mention that viewing the map $Para : Phrase \to Paraphrase$ as a random (discrete) dynamical system may lead to interesting research directions.

4.5.2 Heuristics

Our aim for the remaining of this chapter becomes twofold. First, introduce a heuristic for approximating (4.45). Given sentence S we turn to a collection of sentences C each of which is a sentence-level paraphrase of S. For each sentence $S' \in C$, if S-to-S' IBM-type alignments are known, then a finer (i.e., phrase-level) paraphrase

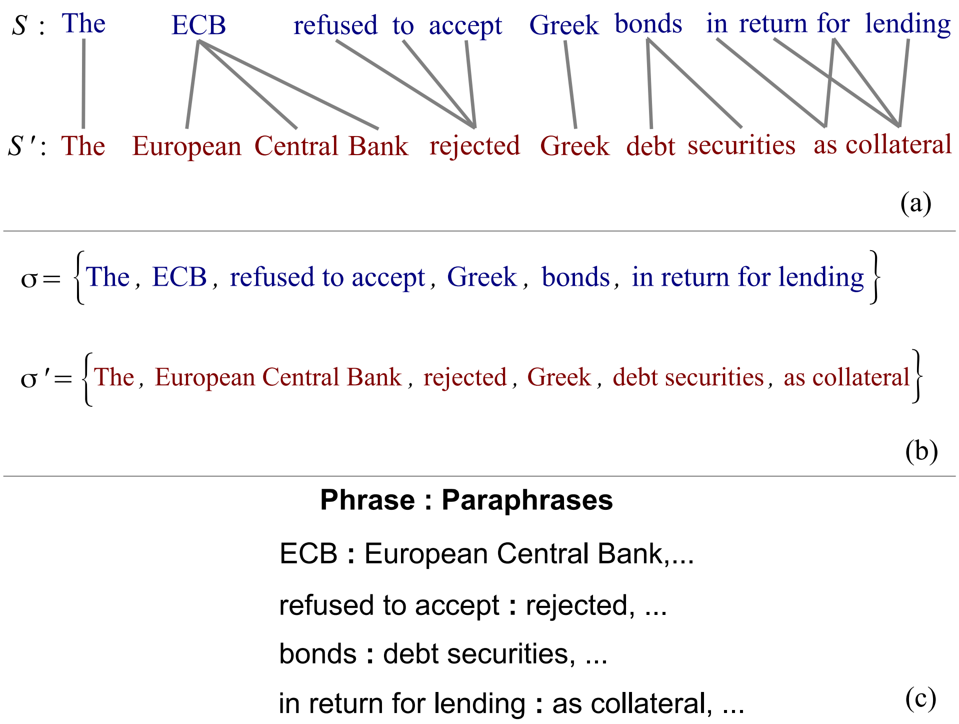

S : The ECB refused to accept Greek bonds in return for lending
S ′: The European Central Bank rejected Greek debt securities as collateral
(a)
σ = {The , ECB , refused to accept , Greek , bonds , in return for lending}
σ ′= {The , European Central Bank , rejected , Greek , debt securities , as collateral}
(b)
Phrase : Paraphrases
ECB : European Central Bank,...
refused to accept : rejected, ...
bonds : debt securities, ...
in return for lending : as collateral, ...
(c)

Figure 4.7: (a) Word alignments between a sentence and its sentence-level paraphrase. (b) Segmentations resulting from (a). (c) Part of the phrase-level dictionary resulting from (a).

dictionary can be constructed: Figure 4.7(a) shows an example of word alignments between sentence S and a possible sentence-level paraphrase S'. As in the bilingual case, such alignments give rise to components, i.e., they yield a partition of words in both S and S'; Figure 4.7(b) shows segmentations σ and σ' for S and S' respectively, which can can be inferred from the partition. Figure 4.7(c) shows a sample from the phrase-paraphrase dictionary that is constructed for each segment in σ: If $x \in \sigma$ is aligned with $x' \in \sigma'$, then x' is considered a paraphrase of x. The resulting full phrase-paraphrase dictionary is of high quality provided that a) the collection C contains high quality sentence-level paraphrases for S, and b) accurate alignments between S and any $S' \in C$ can be obtained.

Sentence S in the example of Figure 4.7 is a simple sentence, i.e., it does not contain a subordinate clause. This property permits a clear distinction between subject and predicate in a sentence; further clauses that bind with the subject or the predicate in a syntactically more complex way are absent. It is thus natural to assume that such sentences are easier to be segmented, paraphrased, etc. Indeed, there is evidence that documents that consist mostly of simple sentences are easier to be summarized [168].

For our purposes, if S is a simple sentence then a high quality phrase-level dictio-

[2] https://oeis.org/A000110

nary is easier to be constructed (because the two conditions stated above are easier to be met). Additionally, the resulting segmentation σ of S as in Figure 4.7(b) is likely to have little variation across the sentences of C. Since the phrase-level dictionary is assumed to be of high quality, then we can deduce that $d(S, O_\rho) \approx 0$ for most paraphrased segmentations $\rho \in P$. Thus, the problem of finding the natural segmentation of S reduces to finding the most frequent segmentation of S, as determined by S's alignments with C's sentences. In fact, if S is not a simple sentence, then the same argumentation holds. The only difference is that C should be sufficiently larger in order to yield the dominant segmentation.

This systematic approach of finding the natural segmentation of a sentence masks potentially deeper understanding of how the core constituents of a sentence are integrated with the sentence itself. Also, even if the phrase-level dictionary is perfect, then this process does not explicitly guarantee that the dominant segmentation is also the natural segmentation, as given by (4.45). Again, without going into details, Birkhoff's Ergodic Theorem from dynamical systems is used as a compass to validate the above process. However, constructing or finding a high quality sentence-level paraphrase collection C is very difficult in practice. This is because existing English-to-paraphrased English corpora are scarce and their construction is a difficult problem in itself.

To this end we turn to a further approximation. Given English sentence S, instead of identifying a sentence-level paraphrased collection C, we seek translations of S in various languages. Since the rest of the process remains the same, there is no technical restriction that prevents the augmentation of the English dictionary with words from other languages. A translation of an English sentence S can thus play the role of a sentence-level paraphrase of S.

In practice, the European parliamentary proceedings (Europarl corpus) can provide the desired collection C of translations for an English sentence S, together with their alignments. Unfortunately S has to be part of Europarl as well and not just a random English sentence. This brings us the second part of our goal: Find a method that approximates (4.45) for any sentence in any language. We tackle this problem using the two segmentation methods that were introduced in Sections 4.2 and 4.3. They are both language independent and can be trained on the same monolingual corpus.

Thus, given sentence S, optimal segmentations σ^*_{VLM} and σ^*_{REF} that result from varied n-gram LMs and segmentation refinements respectively can be compared fairly with each other. If S's natural segmentation σ^* is also known, then all three can be compared to each other and decide whether σ^*_{VLM} or σ^*_{REF} is closer to σ^*.

Using the approximation described above, 250K English sentences from the Europarl corpus can be naturally segmented. In the next section we provide evidence that segmentation refinements are better than varied n-gram LMs in generating natural segmentations.

Family	**Languages**	
Baltic	Latvian, Lithuanian	2
Finno-Ugric	Estonian, Finnish, Hungarian	3
Germanic	Dutch, German, Swedish	3
Greek	Greek	1
Romanic	French, Italian, Spanish	3
Slavic	Czech, Polish, Slovak	3
Total # translations for an English sentence		15

Table 4.1: Languages and their corresponding linguistic families. An English sentence has a translation in each of these languages.

4.5.3 Experiments

We are interested in forming a set of English sentences with each such sentence having a sufficient number of translations. Sufficiency is determined by the ability to yield the dominant segmentation for each sentence in the set. To this end we turn to the Europarl corpus [83] which consists of European parliamentary proceedings: speeches of MEPs are transcribed and translated into all official EU languages.

From release 7 of Europarl[3] we extracted 264,528 English sentences and each such sentence has a translation into 15 other European languages. Table 4.1 shows these languages as well as their linguistic family classification. The size of the set of English sentences and the total number of translations for each English sentence together with the fair distribution across linguistic families, provides a potentially reliable dataset for our experiments.

We do not directly investigate how many translations are sufficient for a linguistically simple or complex English sentence in order to generate its dominant segmentation. Instead, we simply aim for major discrepancies in appropriate experimental measurements that, in retrospect, deem this investigation unnecessary. Similarly for the chosen distribution of languages across their linguistic families. In detail, the experimental process is as follows.

1. For each language L of Table 4.1 the corresponding EN–L Europarl parallel corpus is used to generate word alignments with the standard SMT machinery.

2. We identify all English sentences of those corpora that have translations in all 15 languages of Table 4.1: These form a set, say $\mathbf{S}$, of 264,528 English sentences; each such sentence has a translation in each language of $\mathcal{L} = \{\text{Latvian}, ..., \text{Slovak}\}$.

3. For any sentence $S \in \mathbf{S}$, the collection of translations $\{T_L(S)\}_{L \in \mathcal{L}}$ plays the role of

[3] http://www.statmt.org/europarl/index.html

the collection C that was discussed in the previous section. Equipped with the known word alignments between any $S \in \mathbf{S}$ and each translation of $\{T_L(S)\}_{L\in\mathcal{L}}$ we proceed with finding the most dominant, and thus natural, segmentation for S.

This is determined with the aid of segmentation-bitstring correspondence that was discussed in Section 4.4, together with the heuristic of Section 4.5.2. Given sentence with n words, say $S = s_1^n$, we rewrite a segmentation σ of S as

$$\sigma \;=\; s_1\, b_1\, s_2\, b_2\, s_3\; ...\; s_{n-1}\, b_{n-1}\, s_n, \tag{4.47}$$

where each b_i is '0' or '1', i.e.,

$$b_i = \begin{cases} 1, & \text{if } \; s_i\; s_{i+1} \; \text{ is part of a segment in } \sigma; \\ 0, & \text{otherwise,} \end{cases} \tag{4.48}$$

for all $i = 1, ..., n-1$. For instance, in the example of Figure 4.7 (b) segmentation σ is rewritten as

$$\sigma \;=\; \text{The } \mathbf{0} \text{ ECB } \mathbf{0} \text{ refused } \mathbf{1} \text{ to } \mathbf{0} \text{ accept } \mathbf{0} \text{ Greek } \mathbf{0} \text{ bonds } \mathbf{0} \text{ in } \mathbf{1} \text{ return } \mathbf{1} \text{ for } \mathbf{1} \text{ lending.} \tag{4.49}$$

As mentioned in Section 4.5.2, it is possible to construct a segmentation for S from each translation in $\{T_L(S)\}_{L\in\mathcal{L}}$, i.e., we have

$$\sigma_L \;=\; s_1\, b_{L,1}\, s_2\, b_{L,2}\, s_3\; ...\; s_{n-1}\, b_{L,n-1}\, s_n, \tag{4.50}$$

with

$$b_{L,i} = \begin{cases} 1, & \text{if } \; s_i\; s_{i+1} \; \text{ is part of a segment in } \sigma_L; \\ 0, & \text{otherwise,} \end{cases} \tag{4.51}$$

for all $i = 1, ..., n-1$ and for all $L \in \mathcal{L}$. Finding the dominant, and thus natural segmentation σ^* is a matter of counting how often a '1' appears at bit position i of the 15 segmentations, for all $i = 1, ..., n-1$. Surely, for a given bit position, we cannot expect the same bit to appear in all 15 segmentations. The dominance of bit '1' in a bit position is characterized by the relative frequency of '1's and parametrized by some $\theta \in [0, 1]$. More precisely, for the parametrized dominant segmentation σ_θ^* we have

$$\sigma_\theta^* \;=\; s_1\, b_1^\theta\, s_2\, b_2^\theta\, s_3\; ...\; s_{n-1}\, b_{n-1}^\theta\, s_n, \tag{4.52}$$

with

$$b_i^\theta = \begin{cases} 1, & \text{if } \;\; |\{L \,:\, L \in \mathcal{L},\; b_{L,i} = 1\}| \,/\, 15 \;\geq\; \theta; \\ \\ 0, & \text{otherwise,} \end{cases} \tag{4.53}$$

where $b_{L,i}$ as in (4.51), for all $i = 1, ..., n-1$. I.e., θ tells us how tolerant we want to be when joining two consecutive words in a segment of the dominant segmentation σ_θ^*. If $\theta = 0$, then at least one translation with $b_{L,i} = 1$ at position i for some L will suffice in joining words s_i and s_{i+1} in the same segment in σ_0^*. On the other hand, if $\theta = 1$, then full agreement among all translations is required for the same event to occur in σ_1^*.

In this way the parametrized dominant segmentation for all sentences in **S** is determined. Our ultimate goal, however, is the investigation of whether varied n-gram LMs or refinement-based optimal segmentations are closer to generating natural segmentations. The value of θ will be determined based on that investigation.

4. For each $S \in \mathbf{S}$

- Varied n-gram LM optimal segmentation $\sigma^*_{\texttt{VLM}}$ is determined by (4.10), (4.12) and (4.14).

- Refinement-based optimal segmentation $\sigma^*_{\texttt{REF}}$ is determined by (4.28), (4.30)-(4.35).

For both methods the corpus from which counts are extracted and contribute to (4.10) and (4.28) is a subset W of the fifth edition of the English Gigaword corpus.[4] In particular, W consists of the following news-related resources:

- Agence France-Presse, English service (afp_eng);

- Associated Press Worldstream, English service (apw_eng);

- Los Angeles Times/Washington Post Newswire Service (ltw_eng),

which amount to 72.8M sentences or 2.06B tokens.

As mentioned in Sections 4.2 and 4.3.3 only counts of n-grams with $n = 1, ..., 7$ are collected. Consequently, only segments that consist of at most 7 words may appear in $\sigma^*_{\texttt{VLM}}$ and $\sigma^*_{\texttt{REF}}$. Also, each sentence of W is augmented by the beginning and end-of-sentence tokens, which are treated as normal tokens. Both optimization problems (4.14) and (4.35) are solved with the algorithm of the Cross-Entropy method, as described in Section 4.4.

5. For each $S \in \mathbf{S}$ we determine whether $\sigma^*_{\texttt{VLM}}$ or $\sigma^*_{\texttt{REF}}$ is closer to σ^*_θ using the metric 'Precision' from Information Retrieval. In particular, since σ^*_θ is parametrized by θ, the parametrized precision for each method is given by

$$\text{Prec}_\theta(S, m) \;=\; \frac{|\sigma^*_m \cap \sigma^*_\theta|}{|\sigma^*_\theta|}, \qquad m = \texttt{VLM}, \texttt{REF}. \tag{4.54}$$

The value of θ for which (4.54) is maximized for each method independently, namely

$$\theta^* \;\equiv\; \theta^*(S, m) \;=\; \arg\max_{\theta \in [0,1]} \text{Prec}_\theta(S, m), \qquad m = \texttt{VLM}, \texttt{REF} \tag{4.55}$$

[4] https://catalog.ldc.upenn.edu/LDC2011T07

	VLM	REF
Prec	19.4%	60.7%
θ^*	16.1%	34.9%
PPL	1.0186	1.0218

Table 4.2: Results for Prec(m), $\theta^*(m)$ and PPL(m), m=VLM, REF.

gives rise to the desired metric, i.e.,

$$\text{Prec}(S, m) \;=\; \text{Prec}_{\theta^*}(S, m), \qquad m = \texttt{VLM}, \texttt{REF}. \tag{4.56}$$

In practice (4.55) is obtained by sweeping θ through $[0, 1]$ for each method independently. Finally, the mean precision and mean θ^* over the whole set S are calculated as

$$\text{Prec}(m) \;=\; \frac{1}{|\mathbf{S}|} \sum_{S \in \mathbf{S}} \text{Prec}(S, m), \qquad m = \texttt{VLM}, \texttt{REF} \tag{4.57}$$

and

$$\theta^*(m) \;=\; \frac{1}{|\mathbf{S}|} \sum_{S \in \mathbf{S}} \theta^*(S, m), \qquad m = \texttt{VLM}, \texttt{REF} \tag{4.58}$$

respectively. We are also interested in the perplexity of S under each segmentation method. It is computed in the same way as in Section 4.2:

$$\text{PPL}(m) \;=\; 2^{-\frac{1}{M} \sum_{S \in \mathbf{S}} \ \ell_S(\sigma^*_m)}, \qquad m = \texttt{VLM}, \texttt{REF}, \tag{4.59}$$

where M is the total number of tokens in S and $\ell_C(\omega)$ denotes the log-likelihood of segmentation ω of C, as computed by (4.10) and (4.12).

The results of the above experimental process are reported in Table 4.2. The first, second and third row shows the resulting value of (4.57), (4.58) and (4.59) respectively. In particular, from the first row we conclude that REF is better than VLM in simulating natural segmentations. From the second row we deduce the following: For VLM we have to be lenient when joining two consecutive words in a segment of a dominant segmentation, with an average of only $16.1\% \times 15 = 2.4$ translations to agree at each bit position. This choice for θ, i.e. 0.161, in (4.55) is required on average in order to maximize precision; given that the resulting Prec is only 19.4% we conclude that VLM is a much different type of segmentation method than natural segmentation. On the other hand, REF requires an agreement of $34.9\% \times 15 = 5.2$ translations on average in order to yield a Prec of 60.7%. Although the results for REF are encouraging, we cannot be certain whether it is a method that simulates natural segmentations; inference could have been more robust if linguistically diverse translations (a balance of Indo-European and other family languages) were provided. The third row shows that perplexity under VLM is lower than under REF, which is expected. This is simply because VLM is based entirely on minimizing the perplexity of the corpus that is to be segmented.

Some intuition regarding the type of segmentations that are generated by VLM and REF can be gained by examining the output. In what follows, a sample of sentences from the Europarl corpus is segmented using both methods. For each sentence, the first segmentation output (prefixed by ▷) is computed by applying REF, and the second one (◁) by applying VLM. Two colours, black and red, are used in all outputs to distinguish segments: Each segment is formed by consecutive words that share the same colour.

1. ▷ as far as i am concerned , parliament is a crucial partner in this .

1. ◁ as far as i am concerned , parliament is a crucial partner in this .

2. ▷ bearing in mind the climate change we are facing , i support the introduction of instruments for monitoring the environmental factors in each region separately , along with the allocation of an adequate budget for this .

2. ◁ bearing in mind the climate change we are facing , i support the introduction of instruments for monitoring the environmental factors in each region separately , along with the allocation of an adequate budget for this .

3. ▷ canada has agreed to resolve the difference in exchange for a bilateral regulatory dialogue on biotechnology issues .

3. ◁ canada has agreed to resolve the difference in exchange for a bilateral regulatory dialogue on biotechnology issues .

4. ▷ coal - fired power plants , which have a 40 - year life span and which emit huge amounts of co2 .

4. ◁ coal - fired power plants , which have a 40 - year life span and which emit huge amounts of co2 .

5. ▷ disarmament , arms control and a possible anti - missile shield are also on the agenda .

5. ◁ disarmament , arms control and a possible anti - missile shield are also on the agenda .

6. ▷ everyone is fully aware that the price of energy will increase in future , and europe is currently suffering from a lack of competitiveness in a global world .

6. ◁ everyone is fully aware that the price of energy will increase in future , and europe is currently suffering from a lack of competitiveness in a global world .

7. ▷ for an industry such as the construction sector , it is imperative that the rules are clear and i look forward to seeing the difference these particular changes will make .

7. ◁ for an industry such as the construction sector , it is imperative that the rules are clear and i look forward to seeing the difference these particular changes will make .

8. ▷ freedom of short - term travel is a vital part of preparation for that .

8. ◁ freedom of short - term travel is a vital part of preparation for that .

9. ▷ he was a vice - president and director of us investment bank goldman sachs , where his responsibilities included europe and contact with the national governments there .
9. ◁ he was a vice - president and director of us investment bank goldman sachs , where his responsibilities included europe and contact with the national governments there .

10. ▷ hence , i welcome the comments made by the united nations secretary - general , ban ki - moon , who stated that he will convene a high - level meeting in vienna on 24 june .
10. ◁ hence , i welcome the comments made by the united nations secretary - general , ban ki - moon , who stated that he will convene a high - level meeting in vienna on 24 june .

11. ▷ hiv / aids infections are a global phenomenon and are nothing to do with either so - called risk groups or specific regions .
11. ◁ hiv / aids infections are a global phenomenon and are nothing to do with either so - called risk groups or specific regions .

12. ▷ however , during and after the olympic games , we , unfortunately , had to admit that there was no improvement , but rather a worsening of the human rights situation .
12. ◁ however , during and after the olympic games , we , unfortunately , had to admit that there was no improvement , but rather a worsening of the human rights situation .

13. ▷ i am pleased that among the first to come forward with voluntary offers of assistance to the affected citizens were rescue specialists and volunteer firemen from slovakia .
13. ◁ i am pleased that among the first to come forward with voluntary offers of assistance to the affected citizens were rescue specialists and volunteer firemen from slovakia .

14. ▷ i really do not understand why on the one hand we have to charge full speed ahead without worrying about the impact , whereas on the other hand we move slowly and take the time to think , while , in the meantime , the people are paying the price .
14. ◁ i really do not understand why on the one hand we have to charge full speed ahead without worrying about the impact , whereas on the other hand we move slowly and take the time to think , while , in the meantime , the people are paying the price .

15. ▷ in spite of this , whether or not we want to , we must deal with russia as a partner .
15. ◁ in spite of this , whether or not we want to , we must deal with russia as a partner .

16. ▷ in view of the fact that combating climate change affects competitiveness , our resolution calls for all industrial sectors to be made aware of the danger of carbon leakage and for an end to subsidies on fossil fuels , particularly tax exemptions for the aviation industry .
16. ◁ in view of the fact that combating climate change affects competitiveness , our resolution calls for all industrial sectors to be made aware of the danger of carbon leakage and for an end to subsidies on fossil fuels , particularly tax exemptions for the aviation industry .

17. ▷ it is alarming that the eu has been caught by surprise when massive popular demonstrations against the previous authoritarian regime started .
17. ◁ it is alarming that the eu has been caught by surprise when massive popular demonstra-

tions against the previous authoritarian regime started .

18. ▷ it is also necessary to stimulate investment in small and medium - sized airports to provide increasing interconnections between the different european regions and cities .
18. ◁ it is also necessary to stimulate investment in small and medium - sized airports to provide increasing interconnections between the different european regions and cities .

19. ▷ it should be the member states that decide the extent of mother tongue teaching .
19. ◁ it should be the member states that decide the extent of mother tongue teaching .

20. ▷ let us also hope that the usa and china play an important role in the nuclear disarmament of the korean peninsula .
20. ◁ let us also hope that the usa and china play an important role in the nuclear disarmament of the korean peninsula .

21. ▷ let me take the opportunity to explain why i voted against the greens and their amendment .
21. ◁ let me take the opportunity to explain why i voted against the greens and their amendment .

22. ▷ let us not forget that foot - and - mouth disease is still a serious problem in brazil .
22. ◁ let us not forget that foot - and - mouth disease is still a serious problem in brazil .

23. ▷ marking this 20th anniversary of the release of nelson mandela from prison , jerzy buzek , the president of the european parliament , said :
23. ◁ marking this 20th anniversary of the release of nelson mandela from prison , jerzy buzek , the president of the european parliament , said :

24. ▷ millions of people in europe are afraid of losing their jobs and , frankly , i do not understand the position of the french president in opposing the summit :
24. ◁ millions of people in europe are afraid of losing their jobs and , frankly , i do not understand the position of the french president in opposing the summit :

25. ▷ moreover , the commission should follow up on the trade agreements with the eu 's partner countries by carrying out , prior to and after the signing of a trade agreement , sustainability impact assessment studies , taking into account vulnerable sectors in particular .
25. ◁ moreover , the commission should follow up on the trade agreements with the eu 's partner countries by carrying out , prior to and after the signing of a trade agreement , sustainability impact assessment studies , taking into account vulnerable sectors in particular .

26. ▷ mortality rates while waiting for a transplant usually range from 15 % to 30 % .
26. ◁ mortality rates while waiting for a transplant usually range from 15 % to 30 % .

27. ▷ part of the hizbollah 's raison d'être is the ongoing occupation by israel of the shebaa farms .

27. ◁ part of the hizbollah ’s raison d’être is the ongoing occupation by israel of the shebaa farms .

28. ▷ peace and security in kosovo ought to be a priority for the union , since it neighbours the former yugoslav republic of macedonia , a candidate country to the eu , and we should play a leading role as a mediator between serbia and kosovo .
28. ◁ peace and security in kosovo ought to be a priority for the union , since it neighbours the former yugoslav republic of macedonia , a candidate country to the eu , and we should play a leading role as a mediator between serbia and kosovo .

29. ▷ similarly , we must take into account the need to recognise the specific nature of natural disasters caused by droughts and fires in the mediterranean region and adapt our prevention , investigation , risk management , civil protection and solidarity mechanisms accordingly .
29. ◁ similarly , we must take into account the need to recognise the specific nature of natural disasters caused by droughts and fires in the mediterranean region and adapt our prevention , investigation , risk management , civil protection and solidarity mechanisms accordingly .

30. ▷ some people do not like you because you are too european , and others because of ideological prejudice .
30. ◁ some people do not like you because you are too european , and others because of ideological prejudice .

31. ▷ the need for one single internal market is crucial for europe to take a leadership in the global economy , with a focus on the service sector and the knowledge economy .
31. ◁ the need for one single internal market is crucial for europe to take a leadership in the global economy , with a focus on the service sector and the knowledge economy .

32. ▷ the eu - russia summit coincides with a crucial time in the history of russia , namely the end of mr putin ’s era .
32. ◁ the eu - russia summit coincides with a crucial time in the history of russia , namely the end of mr putin ’s era .

For the output generated by REF (▷), segments generally fall under the following categories:

1. Idiom. Examples include *bearing in mind* (in sentence 2); *in spite of* (15); *caught by surprise* (17); *taking into account* (25).

2. Adjective phrase, i.e., a phrase containing an adjective that modifies a noun phrase. Also followed by a preposition or a conjunction. Examples include *huge amounts of* (4); *short - term* (8); *fully aware that* (6); *small and medium - sized* (18).

3. Prepositional verb, i.e., a verb followed by a preposition. Also preceded by an auxiliary verb due to conjugation. Examples include *has agreed to* (3); *deal with* (15); *voted against* (21); *follow up on* (25).

4. Proper noun. Also preceded by a determiner. Examples include *the eu* (17); *the greens* (21); *jerzy buzek* (23); *former yugoslav republic of macedonia* (28).

5. Noun phrase. Also preceded by a determiner, and/or followed by a preposition. Examples include *life span* (4); *foot - and - mouth disease* (22); *the signing of a* (25); *nature of* (29).

6. Conjunction, conjunctive adverb and subordinating conjunction. Also followed by a comma. Examples include *however ,* (12); *on the one hand ... on the other hand* (14); *whether or not* (15); *similarly ,* (29).

7. Clause or incomplete clause (lacking subject and/or object). Also followed by a subordinating conjunction. Examples include *support the introduction of* (2); *it is imperative that* (7); *i am pleased that* (13); *do not understand why* (14); *resolution calls for* (16); *to explain why* (21); *is crucial for* (31).

Also, again for REF, the following observations can be made.

i) Segments are formed on the basis of statistics rather than syntax. I.e., segments are "common" groups of consecutive words rather than children of the same parent node of a parse tree.

ii) "Commonness" in i) is determined by whether REF characterizes a phrase as frequent enough in the training data. Our training data (Step 4 in the experimental process) consist entirely of news resources. This explains the coarse grained segmentation of certain noun phrases. Examples include *coal - fired power plants* instead of *coal - fired power plants* (5); *korean peninsula* instead of *korean peninsula* (20); *serbia and kosovo* instead of *serbia and kosovo* (28).

iii) Words are not forced to be part of large segments. In fact, segments consisting of unigrams are prevalent in all sentences; they also fall under the categories listed above.

iv) *Segments are of joint maximal expressive power and minimal length within a given segmented sentence*. Let s and $\bar{s}$ denote sequences of words; given a segmented sentence, the operation of splitting segment $x = (s\bar{s})$ into new segments (s) and $(\bar{s})$ results in the following: At least either (s) or $(\bar{s})$ fails to provide succinctly its grammatical/syntactic/semantic role in the sentence; the grouping of s and $\bar{s}$ under one segment reduces their stand-alone ambiguity. This holds regardless of how s and $\bar{s}$ are chosen within segment $(s\bar{s})$, i.e., splitting segment x always results in increasing ambiguity. On the other hand, let y denote the segment adjacent to the left (right) of x. Then, the formation of a new segment yx (xy) does not provide further insight into the role of x and y in the sentence.

The last observation above is equivalent to the abstract semantic decomposition that was discussed in Section 4.1: Namely the semantic "linear combination" of minimal segments that matches the meaning of the sentence. More precisely, let $\sigma^*_{\texttt{REF}}$

$= \{x_1, x_2, ..., x_k\}$ denote a segmentation of a given sentence. If M is an abstract operator that maps phrases to meaning, and if $\oplus$ is a semantic binary operation, we have

$$M(x_1 x_2 ... x_k) = M(x_1) \oplus M(x_2) \oplus ... \oplus M(x_k), \tag{4.60}$$

i.e., the meaning of the sentence is linearly composed by the meaning of the segments generated by REF. Also, segmentation $\sigma^*_{\texttt{REF}}$ is minimal in the following sense: For all $x \in \sigma^*_{\texttt{REF}}$ and for all phrases $s, \bar{s}$ such that $x = (s\bar{s})$, we have

$$M(x) = M(s\bar{s}) \neq M(s) \oplus M(\bar{s}), \tag{4.61}$$

i.e., further splitting of the segments results in loss of linearity.

The above is also closely related to the formal definition of natural segmentation as presented in (4.45). Suppose that sentence S is segmented using REF. As mentioned in observation iv) above, the resulting segments have a stand-alone function. Thus, for each such segment x, its paraphrases are less likely to deviate from x's function in S. The worst case paraphrased segmentation, say r, gives rise to the sentence-level paraphrase O_r. In its turn, O_r is expected not to deviate much from S. More accurately, it is likely that $d(S, O_r)$ will be *relatively* small. By taking into account observation i) above and (4.61) we also have that segments in $\sigma^*_{\texttt{REF}}$ are both frequent enough and minimal in length. The former suggests that $\sigma^*_{\texttt{REF}}$'s segments can pass strict likelihood tests. The latter strengthens the claim that $d(S, O_r)$ will be *relatively* small; any other segmentation that passes strict likelihood tests and generates worst case sentence-level paraphrase $O_{r'}$, is bound to satisfy $d(S, O_{r'}) > d(S, O_r)$. Hence, we conclude that the assertion $\sigma^* \approx \sigma^*_{\texttt{REF}}$ is valid.

Segments generated by VLM (◁) also fall under the above categories, but are generally larger than REF. Observations i)–iii) above also hold for VLM, but observation iv) does not. Therefore, it cannot be suggested that VLM simulates natural segmentations.

Throughout the experimental process, the maximum number of words that is allowed to form a segment in $\sigma^*_{\texttt{REF}}$ and $\sigma^*_{\texttt{VLM}}$ is 7. For completeness, the same process is repeated with the maximum such value ranging from 3 to 6. Results for Prec, θ^* and PPL are shown in Figures 4.8, 4.9 and 4.10 respectively; the results from Table 4.2 are also included to facilitate comparisons. For both methods, these quantities are monotone decreasing in maximum segment length, with the exception of θ^* for VLM. The smaller the maximum segment length, the larger the values for Prec, θ^* and PPL. This is a positive result for the following reason: It implies that segments generated by the heuristic method for natural segmentation are *smaller* than the ones generated by REF, but of *similar type* to the ones generated by REF. It thus strengthens the assertion that $\sigma^* \approx \sigma^*_{\texttt{REF}}$.

4.6 Related work

The varied n-gram LMs that were introduced in Section 4.2 do not construct models during the training stage, which is basically the counting of all k-grams, for all $k \leq n$,

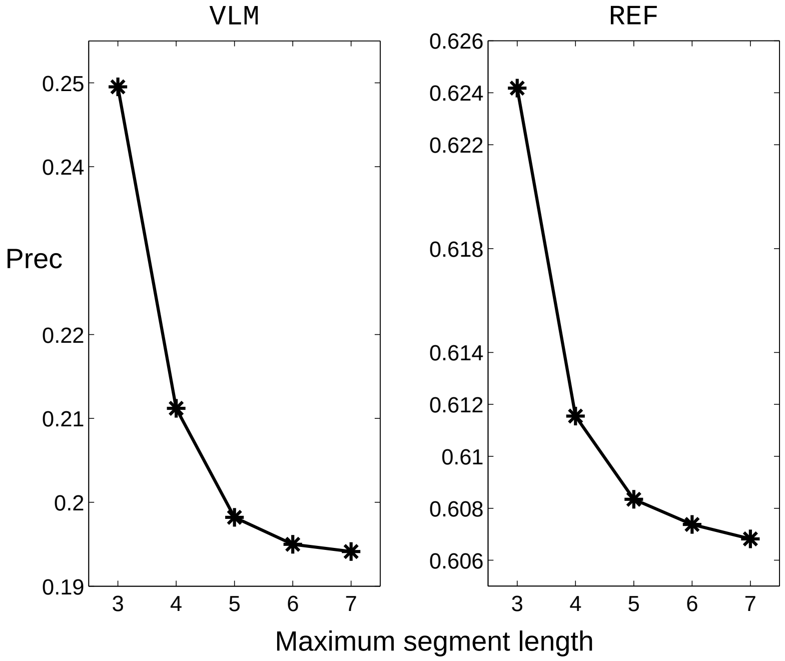


Figure 4.8: Prec as a function of maximum segment length for methods `VLM` and `REF`.

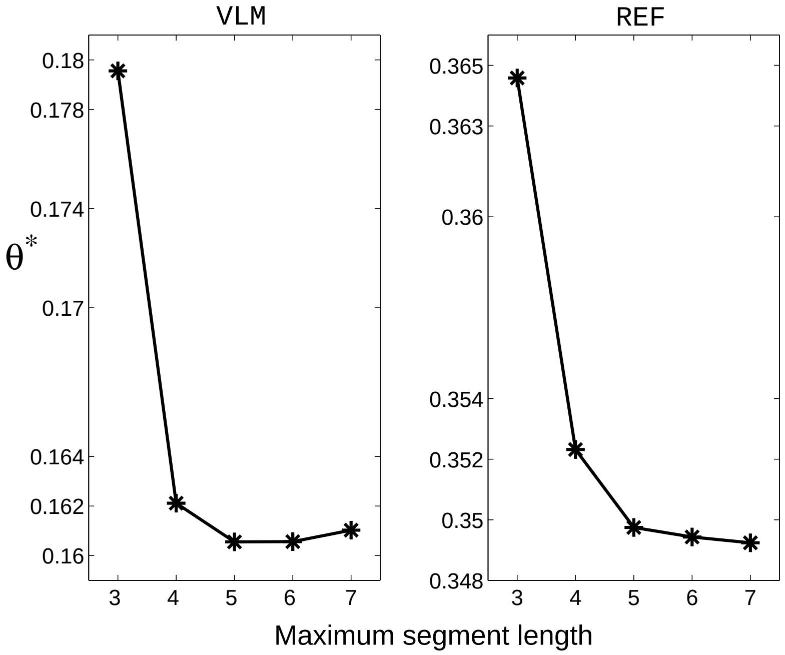


Figure 4.9: θ^* as a function of maximum segment length for methods `VLM` and `REF`.

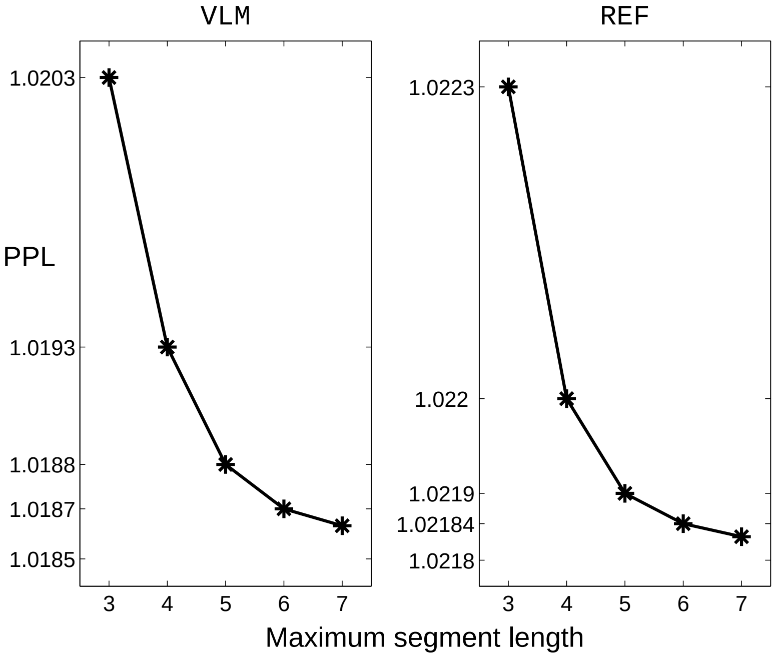


Figure 4.10: PPL as a function of maximum segment length for methods VLM and REF.

from a large (training) corpus. Based on those counts, for a test corpus, each of its words ‘chooses’ its own history, which essentially results in the optimal segmentation of the test corpus. Our method should not be confused with other methods that construct models of varied history at training stage, i.e., by backing off to smaller n-grams if a likelihood-based criterion dictates so [80, 98, 124, 133, 139, 143].

For the task of query segmentation in Information Retrieval, Pointwise Mutual Information is traditionally used as a baseline [64, 66, 75, 103, 125, 145]. However, other than empirical success, the motivation behind its use is unchallenged. In [137] Shannon’s entropy was axiomatized in the setting of partial orders that is formed by partitions of finite sets. Taking a step forward, Simovici [138] established the relationship between Shannon’s entropy and metrics on lattices that are formed by partitions. In other words, there is sound reasoning why an entropy-like criterion (e.g., Pointwise Mutual Information) can discern finer segmentations of a given phrase. Our work in Section 4.3 complements those findings.

Our motivation behind the criteria of natural segmentation lie mainly in the resulting phrase-tables after pruning [74, 163]; to lesser extent, similar requirements have been posed by other work in SMT that perform monolingual segmentations [13, 116]. It is imperative of future work to establish links of our definition of natural segmentation with canonical composition in formal semantics [68, 90, 147, 161].

Finally, the Cross-Entropy method was used instead of the more conventional dynamic programming algorithms that are used in SMT: The Forward-Backward algorithm [122] has been consistently used for obtaining high-quality alignments [38, 41, 104, 157] and the beam search stack decoding algorithm [151, 152] is one of the standard approaches for efficient searching of candidate translations.

4.7 Conclusions

The focus of this chapter was the monolingual version of the 'component' that was discussed in Chapter 3. This monolingual object is simply a phrase, or segment, within a sentence S; the identification of all S's such segments results in a particular partition for S, termed the natural segmentation of S:

RQ3 *Given a sentence in some language, identify what conditions a segmentation of the sentence should satisfy, in order for linear compositionality of meaning to hold. How can one define the segmentation that satisfies those conditions optimally?*

The definition of the natural segmentation of a sentence revolved around paraphrases. In lay terms, the natural segmentation of a sentence is the smallest grouping of its words that forms a syntactic/semantic backbone for that sentence. A segment x of a naturally segmented sentence should be 1) Easily replaced by a paraphrase: Substituting x with any of its paraphrases should not cause 'much' syntactic/semantic ambiguity in the sentence. 2) Frequent 'enough' in large corpora.

The exact definition, as presented in Section 4.5.1, requires a sentence-level semantic metric. This abstraction makes the formal definition difficult for applications. To this end, we turned to novel statistical segmentation methods that would ideally simulate the output of natural segmentations, but without invoking such metrics.

The first method, termed `VLM`, was a generalization of n-gram language models (LMs), namely varied n-gram LMs. It relied on finding the sufficient memory for predicting a word in a particular sentence. By considering the same problem for each word in that particular sentence, the result is a segmentation of 'minimum perplexity'.

The second method, termed `REF`, dealt with estimating costs for perturbing a segmentation into another:

RQ4 *Given the relationship between Shannon's entropy and metrics on lattices [138], elaborate on the mathematical framework of Pointwise Mutual Information (PMI). Is it possible to extend PMI within this framework for simulating natural segmentations?*

For a particular sentence, the set of all its segmentations together with the operation of 'refinement' (splitting of a segment into two new segments) forms a particular type of partially ordered set, namely a lattice. Choosing a metric on that lattice appropriately, resulted in the formation of cost functions for segments and, consequently, segmentations of that sentence. The optimal segmentation was calculated as the one that is the least expensive to be perturbed into. The development of this method revealed a previously unseen theoretical link for the so-called Pointwise Mutual Information (PMI): It was shown that PMI is a metric on the said lattice.

For a given English sentence, the experimental process required the generation of its natural segmentation and the segmentations produced by VLM and REF. In order to compare these three to each other, some approximations had to be done for generating the natural segmentation. For a given phrase in the sentence, the role of its paraphrases was played by phrase-level translations. These were found from word-aligning the sentence with 15 translations, each from a different language. In general, word-alignments yield a partition for any sentence (as well as for its translation). Thus, those 15 different partitions were used for finding the dominant segmentation for the English sentence, which eventually operated as the approximation of the natural segmentation.

Examination of the output showed that REF's segmentation of a given sentence met the desired criteria: Its segments had a minimal, unambiguous stand-alone function in the sentence. This observation, combined with results from the experimental process, showed that REF was found to be better than VLM in simulating natural segmentations.

Generating output of all methods discussed in this chapter required searching through all possible segmentations of a given sentence. This was done efficiently with a novel application of the Cross-Entropy method for combinatorial optimization. It is explained in Chapter 5 how this method can be easily adapted to the more general problem of extracting optimal bilingual segmentations.

Natural language processing branches, such as composition in distributional semantics [8, 28], could benefit from the findings of this chapter. In SMT, a possible application of natural segmentations could be via the assesment of the N-best list of candidate translations during decoding. This means that candidate translations with higher natural segmentation scores should also be ranked higher as potential translations. In Chapter 5 the notion of natural segmentation is extended to the bilingual level and direct applications to SMT are investigated.

Chapter 5

Bilingual Segmentations

This chapter is devoted to answering **RQ5** and **RQ6**. The aim of this chapter is to generalize the notion of natural segmentation from the monolingual setting of Chapter 4 into a bilingual setting. By bilingual setting we mean the consideration of a pair of sentences which comes from a pair of languages. Furthermore, these sentences are assumed to be translations of each other. By simply identifying the natural segmentation of each sentence (in its corresponding language) independently, one would overlook important structure, thus not achieving anything. This structure is the inherent correspondence between segments of the two sentences. The algorithmic identification of these correspondences is in fact the goal of this chapter. The following two conditions are introduced in order to help us define the bilingual natural segmentation: 1) Segments in a sentence abide to natural segmentation criteria (in its corresponding language), as described in Chapter 4. 2) Segment pairs in the sentence pair respect basic phrase-level dictionary entries of the corresponding language pair. The bilingual natural segmentation is defined as the one that meets these two conditions optimally. The dictionary entries of Condition 2 are assumed to be the word alignments that result from the training stage of Statistical Machine Translation. As described in Chapter 3, each bilingual word-aligned segmentation of a sentence pair has a graph representation. Thus, the identification of the bilingual natural segmentation reduces to forming a partition of connected components of the sentence pair that meets optimally the above conditions. For Condition 1, the quantitative criteria that were introduced in Chapter 4 are carried over intact. The appropriate graph-based interpretation of Condition 2, relates to the degree of difficulty of perturbing components into disconnected graphs. In our experiments, we use the phrase pairs that emerge from connected components of bilingual natural segmentations in order to form the phrase-table. Results show that such phrase pairs form the core of the effective set of translation rules.

5.1 Introduction

In this chapter we extend the notions from the monolingual setting of Chapter 4 to a bilingual setting. It is essentially an attempt to generalize the concept of natural segmentation from a single sentence in any language to any sentence pair with known word alignments. For a given word-aligned sentence pair, an obvious approach for tackling this problem would be to identify the natural segmentation of each sentence of the pair independently. However, as it will be evident later in this chapter, the presence of word alignments should not be ignored. Instead, additional effort should be devoted into achieving such a bilingual natural segmentation:

RQ5 *Given a word-aligned sentence pair, identify what conditions a bilingual segmentation of the pair should satisfy in order to form a bilingual natural segmentation. How can one define the bilingual segmentation that satisfies those conditions optimally?*
Optimally, the following conditions should hold simultaneously:

1. Segments in both source and target language sentences abide to natural segmentation.

2. Source and target language segments are synchronized with each other via word alignments.

Condition 1 is exactly what was discussed in Chapter 4: A natural segmentation of a sentence in any language is a segmentation whose segments provide succinctly their grammatical/syntactic/semantic role in the sentence and they do so minimally. Condition 2, which is the focus of this chapter, is identified as the necessary condition that permits the said generalization. ‘Synchronization via word alignments’ needs to be elaborated further. This is done best by first describing another challenge that is addressed in this chapter.

In Section 3.4 it was explained that a word-aligned sentence pair has the following graph representation: Its source and target language words can be viewed as source and target type vertices respectively; word alignments play the role of edges connecting source and target type vertices. No source-to-source nor target-to-target vertices are assumed and the graph is thus bipartite. The top graph of Figure 5.1 shows an example of such a representation. Word alignments admit a partition in a sentence pair; each part, or component, consists of words from the sentence pair and alignments that connect source words with target words in the following way: i) It is possible to form a path between any two words of the component via word alignments, and ii) It is impossible to form such a path between any word in the component and any word outside the component. In Statistical Machine Translation, the process of word-aligning training data, i.e., a large collection of sentence pairs, is followed by the extraction of translation rules. For a given word-aligned sentence pair, only certain types of phrase

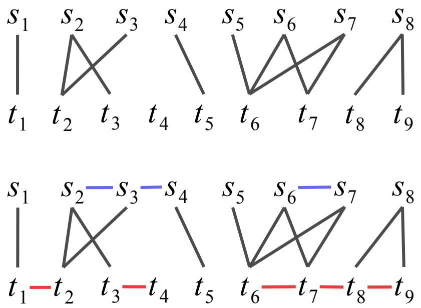

s_1 s_2 s_3 s_4 s_5 s_6 s_7 s_8
t_1 t_2 t_3 t_4 t_5 t_6 t_7 t_8 t_9
s_1 s_2 s_3 s_4 s_5 s_6 s_7 s_8
t_1 t_2 t_3 t_4 t_5 t_6 t_7 t_8 t_9

Figure 5.1: Top: Graph representation of a word-aligned sentence pair (s_1^8, t_1^9). Bottom: One possible bilingual segmentation of the above. Source-to-source and target-to-target edges are shown with blue and red respectively.

pairs are allowed to become translation rules. A phrase pair is a translation rule if and only if the following conditions hold:

- Its words respect the order of appearance in the sentence pair.
- Its words are in one-to-one correspondence with the words of a union of components.

Equivalently, a translation rule is formed by taking an *arbitrary* collection of components, extracting its words and then ordering them in a way so that the resulting phrase pair is a substring of the sentence pair.[1] Let C denote the set of components and let p denote a phrase pair. Then the set of translation rules can be compactly written as

$$P = \{\, p \,:\, p \in \mathcal{P}(C) \,\}, \tag{5.1}$$

where $\mathcal{P}(X)$ denotes the powerset of set X.

Empirically, however, it has been shown that only a small subset of all possible extracted translation rules is actually useful during decoding [74, 163]. In other words, considering all possible unions of components results in superfluous translation rules. Can we identify the qualitative characteristics of the effective set of translation rules? Is it possible to create an algorithm that generates them? In this chapter we claim that translation rules emerging from bilingual natural segmentations is the core of the effective set. A result from Chapter 3 that connects bilingual segmentations with the (full) set of translation rules is useful for validating the claim above.

In Section 3.5 it was explained that the graph representation of a word-aligned sentence pair is just one possible configuration of a more general system. Namely the one that permits source-to-source and target-to-target edges. The bottom graph of Figure 5.1 shows an example of such a configuration. Surely, the top graph of

[1] The source phrase of the phrase pair is a substring of the source sentence of the sentence pair. Similarly for the target side.

Figure 5.1 represents the exceptional case of this general system, namely the one with complete absence of monolingual edges; these are basically different configurations of bilingual word-aligned segmentations. In general, for a given sentence pair, let σ and τ denote a segmentation for the source and target sentence, respectively. Suppose that the sentence pair is word-aligned. Then let $C(\sigma, \tau)$ denote the set of components that is formed by words, word-alignments and edges induced by segmentations. It was shown that the set of extracted translation rules from the word-aligned sentence pair is given by

$$P = \bigcup_{\sigma,\tau} \{ \, p \, : \, p \in C(\sigma, \tau) \, \}, \tag{5.2}$$

where the union is over all possible bilingual segmentations, i.e., configurations. The equation above suggests that, if bilingual segmentations are taken into account, then the effective set of translation rules can be traced in the set of components *only*, (and not arbitrary unions thereof). Indeed, our task is to identify these special configurations that give rise to the effective set of translation rules. To this end, we turn to bilingual natural segmentations:

RQ6 *What is the effect of bilingual natural segmentations on SMT?*

Let (σ_1, τ_1) denote the bilingual segmentation that optimally satisfies Conditions 1 and 2 above. Let $\{(\sigma_i, \tau_i)\}_{i=2}^{N}$ denote $N-1$ bilingual segmentations that are in the vicinity of (σ_1, τ_1). Also, let (σ_0, τ_0) denote the exceptional bilingual segmentation in which no source-to-source nor any target-to-target edges exist (as in the top graph of Figure 5.1). In this chapter, experiments will show that the effective set of translation rules is given by

$$P_{\text{eff}} = \bigcup_{i=0}^{N} \{ \, p \, : \, p \in C(\sigma_i, \tau_i) \, \}, \tag{5.3}$$

where N is not 'too large'. It follows that translation rules of use to SMT have specific structure: Basic phrase pairs, or building blocks of bilingualism (the equivalent of words/tokens in monolingualism), together with larger phrase pairs that have stand-alone or almost stand-alone function in sentence pairs, but are of minimal or almost minimal size. Before we explain how bilingual segmentations are chosen in the vicinity of the bilingual natural segmentation, we first elaborate on Condition 2 above.

The formation of components in a bilingual word-aligned segmentation is again the focus of Condition 2, but from a purely structural point of view. Vertices are no longer representations of words; they are treated as labelled nodes, with labels inherited by the order of the words in the sentences. Figure 5.2 shows such a graph representation for a word-aligned sentence pair under two different configurations. In general, we are interested in assessing how robust, i.e., structurally stable, the components are of a given configuration. A key aspect of components is that they are connected. By assessing how 'difficult' it is for a component to lose its connected status, we thus assess its

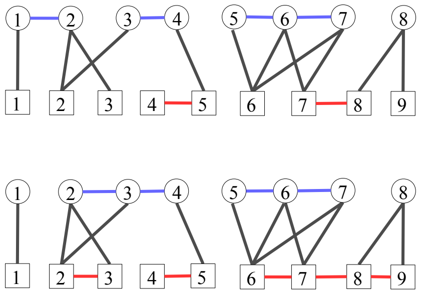


Figure 5.2: Two configurations that overlook words in their representations. Source and target vertices are shown with circles and squares respectively. Source-to-source and target-to-target edges are shown with blue and red respectively.

robustness; by doing so for all components in a configuration, we can draw conclusions about the robustness of the configuration and compare with other configurations.

In this context, robustness of a component is characterized by the extent to which word alignments can be deleted from the component without violating its component status. Informally, the more the required deletions until disconnection, the more robust the component is. Robustness of a component can be succinctly quantified using a well-known graph-theoretic concept, namely the connected spanning subgraph (CSSG). A CSSG of any graph G, is a subgraph H of G such that i) H spans G's vertex set, i.e., all of G's vertices are in H, and ii) H is connected. In other words, a CSSG of G is a subgraph of G that may lack G's edges (but not vertices), while remaining connected. For a particular graph there are several possible CSSGs; the problem of finding the total number of CSSGs is #P–hard [155].

For our purposes however, it is possible to achieve good approximations in linear time. For a given component, its approximated number of CSSGs is then appropriately normalized in $(0, 1]$, thus defining its robustness (with '1' standing for fully robust). High robustness of a component implies that its underlying source and target segments are synchronized well via their word alignments. Finally, a configuration's robustness is given by the geometric mean of its components' robustness. The reason why geometric mean is chosen as the final quality measure is because different configurations lead to different number of components, as in the example of Figure 5.2 (top and bottom graphs have two and three components, respectively).

The extent to which Conditions 1 and 2 above are met by some configuration (σ, τ) with known and fixed word alignments is quantified by a monolingual *surface* and bilingual *structural* measure respectively. We write $g(\sigma, \tau)$ and $f(\sigma, \tau)$ to denote the quality of the surface and structural measure respectively. The value for $f(\sigma, \tau)$ is given by the configuration robustness measure described above. For the surface measure we set $g(\sigma, \tau) = F[g(\sigma), g(\tau)]$, where $F[X, Y]$ is the harmonic mean of X and Y. We experiment with segmentation methods VLM and REF from Chapter 4; the value $g(\rho)$ of segmentation ρ is assessed by both (5.10) and (5.34). The final measure that evaluates the extent to which (σ, τ) jointly meets Conditions 1 and 2 is given by $F[g(\sigma, \tau), f(\sigma, \tau)]$. Thus, the aim of this chapter can be summarized as follows: Given a word-aligned sentence pair, determine its natural bilingual segmentation, as given by

$$(\sigma^*, \tau^*) \;=\; \operatorname*{arg\,max}_{\sigma,\tau} \; F[g(\sigma, \tau), f(\sigma, \tau)], \tag{5.4}$$

where the search is over all possible configurations. Moreover, investigate the effect of (σ^*, τ^*), as well as of other bilingual segmentations in the vicinity of (σ^*, τ^*), on SMT.

The surface and structural measures are incorporated in one algorithm that extracts an N-best list of bilingual word-aligned segmentations. This algorithm, which is an adaptation of the Cross-Entropy method [128], performs joint maximization of surface (in both languages) and structural quality measures. Components of graph representations of the resulting N-best lists give rise to high quality translation rules. These rules, which form a small subset of all possible (continuous) consistent phrase pairs, are used to construct SMT models. Results on Czech–English and German–English datasets show a 90% reduction in phrase-table sizes, which are in line with other pruning techniques in SMT [74, 163]. Experiments also justify the inclusion of the structural measure: By setting $f(\sigma, \tau) = 1$, for all configurations (σ, τ), i.e., by ignoring the presence of word-alignments, translation quality drops. Insignificant loss in translation quality is observed only in the case where the surface measure is powered by `REF` and the structural measure is included.

5.2 Monolingual surface quality measure

Given a sentence in any language, we assess the quality of a segmentation ω of the sentence using the methods introduced in Chapter 4: Varied n-gram language models from Section 4.2 and segmentation refinements from Section 4.3. For brevity, the former is termed `VLM` and the latter is termed `REF`. For the purpose of extracting natural bilingual segmentations, these methods operate in exactly the same way. For completeness, we restate the necessary formulae for assessing the quality of ω based on each method.

Both methods require a large corpus for collecting training data; if a denotes a sequence of words, i.e., phrase, then let count(a) denote the frequency of a in the corpus. Throughout, if a and b are phrases, then their concatenation ab is also a phrase. Segments are phrases and if x is a segment in ω ($x \in \omega$), then let x_j denote the jth word in the sequence of x.

Method `VLM` Let w be a word and let a be a phrase. Consider the probability

$$p(w|a) = \frac{\text{count}(aw)}{\text{count}(a)}, \tag{5.5}$$

if count$(a) \neq 0$, and $p(w|a) = 0$, otherwise. Then the quality of segmentation ω is given by its log-likelihood, i.e.,

$$g(\omega) \; = \; \sum_{x \in \omega} \sum_{j=1}^{|x|} \log p\left(x_j | \, x_1^{j-1}\right), \tag{5.6}$$

where $|a|$ denotes the number of words in segment a.

Method `REF` Let ρ denote a segmentation of the sentence and let a be a segment in ρ. Consider the probability

$$p_\rho(a) = \frac{\text{count}(a)}{\sum_{a' \in \rho} \text{count}(a')}. \tag{5.7}$$

Denote by $\rho \to \rho'$ the pair of segmentations ρ and ρ', such that ρ' is a refinement of ρ. For each such $\rho \to \rho'$ there exists unique pair of phrases $(a, \bar{a})$ such that $a\bar{a} \in \rho$ or $\bar{a}a \in \rho$ and $a, \bar{a} \in \rho'$; this pair is responsible for refining ρ into ρ', i.e., all other segments in ρ remain intact in ρ'. Let

$$R_+(a) \; = \; \{ \, \rho \to \rho' : \; \exists \bar{a} \text{ such that } a\bar{a} \in \rho \text{ and } a, \bar{a} \in \rho' \, \}, \tag{5.8}$$

$$R_-(a) \; = \; \{ \, \rho \to \rho' : \; \exists \bar{a} \text{ such that } \bar{a}a \in \rho \text{ and } \bar{a}, a \in \rho' \, \}, \tag{5.9}$$

denote the sets of refinements for which a appears as a refined segment. Also, let $R(a) = R_+(a) \; \cup \; R_-(a)$. Consider the quantities

$$I_+(a, \rho \to \rho') \; = \; \log \frac{p_{\rho'}(a) \, p_{\rho'}(\bar{a})}{p_\rho(a\bar{a})}, \tag{5.10}$$

$$I_-(a, \rho \to \rho') \; = \; \log \frac{p_{\rho'}(a) \, p_{\rho'}(\bar{a})}{p_\rho(\bar{a}a)}, \tag{5.11}$$

where $\bar{a}$ is the appropriate conjugate phrase in each case. Then the quality of segmentation ω is given by

$$g(\omega) \; = \; \sum_{x \in \omega} \frac{1}{|R(x)|} \left(\sum_{r \in R_+(x)} I_+(x, r) \; + \sum_{r \in R_-(x)} I_-(x, r) \right). \tag{5.12}$$

5.3 Bilingual structural quality measure

Given a word-aligned sentence pair, we introduce a purely structural measure that assesses the quality of its bilingual segmentations. By ‘purely structural’ it is meant that the focus is entirely on combinatorial aspects of the bilingual segmentations and the word alignments. For that reason we turn to a graph theoretic framework.

5.3.1 Connected spanning subgraphs

A segment can also be viewed as a chain, i.e., a graph in which vertices are the segment's words and an edge between two words exists if and only if these words are consecutive. Then, a source segmentation σ and a target segmentation τ are graphs that consist of source chains and target chains respectively. The graph formed by σ, τ and the translation edges induced by word alignments is thus a graph representation of a bilingual word-aligned segmentation.

We focus on a particular type of subgraphs of this representation, namely its connected components, or simply components. A component is a graph such that i) there exists a path between any two of its vertices, and ii) there does not exist a path between a vertex of the component and a vertex outside the component. Condition i) means, both technically and intuitively, that a component is connected and Condition ii) requires connectivity to be maximal.

Components play a key role in SMT. The most widely used strategy for extracting high quality phrase-level translations without linguistic information, namely the consistency method [82, 109] is entirely based on components of word aligned unsegmented sentence, as explained in Chapter 3. In particular, each extracted translation is either a component or the union of components. Since an unsegmented sentence pair is just one possible configuration of all possible bilingual segmentations, we consequently have no direct reason to investigate further than components.

In order to get an intuition of the measure that will be introduced in this section, we begin with an example. Figure 5.3, shows two different configurations of the pair (σ, τ) for the same sentence pair with known and fixed word alignments. Both configurations have the same number of edges that connect source vertices (3) and the same number of edges that connect target vertices (2). However, one would expect the top configuration to represent a better bilingual segmentation. This is because it has more components (4 opposed to 2 for the bottom configuration) and because it consists of components that are of higher structural quality, i.e., more robust clusters.

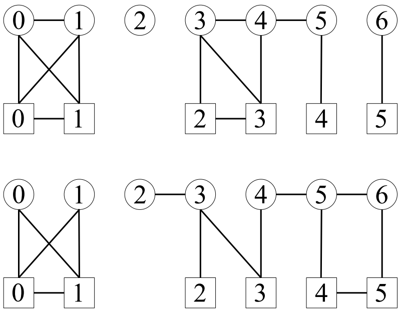


Figure 5.3: Graph representations of two bilingual segmentations with fixed word alignments. Source and target vertices are shown with circles and squares respectively.

A general measure that would capture this observation requires a balance between the number of edges of source and target chains, the number of components and the number of translation edges, all coupled with how these edges and vertices are connected. This might seem as a daunting task that can be tackled with a combination of heuristics, but there is actually a graph-theoretic measure that can fully describe the sought structure. We proceed with introducing this measure.

Let C denote the set of components of the graph representation of a bilingual word-aligned segmentation. We are interested in measuring the extent to which we can delete translation edges from $c \in C$, while retaining its component status. Let a_c denote the subset of translation edges that are restricted to component c. Define the positive integer

$$\begin{aligned} D(c) \;=\; & \text{number of ways of} \\ & \text{deleting translation edges from } a_c, \\ & \text{while keeping } c \text{ connected,} \end{aligned} \tag{5.13}$$

where the option of deleting nothing is counted. Figure 5.4 shows an example of what (5.13) actually counts for a component c:

- One way of deleting nothing (top row, the component itself).
- Three ways of deleting one translation edge (middle row).
- Three ways of deleting two translation edges (bottom row).

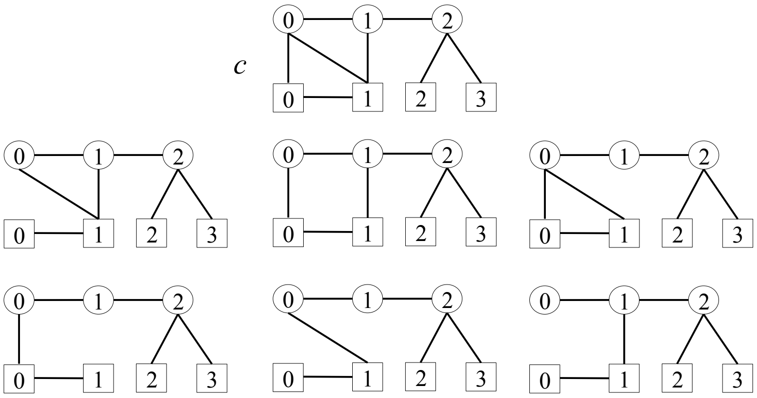


Figure 5.4: Component c (top row) and subgraphs of c that correspond to all possible ways of deleting translation edges without violating connectivity. Source and target vertices are shown with circles and squares respectively.

Other possibilities result in loss of connectivity and we thus have $D(c) = 7$. Observe that each possible way of performing deletions of translation edges while retaining connectivity, corresponds to a subgraph of c. In graph theory, this type of subgraph is known as connected spanning subgraph (CSSG) and is the key quantity of network reliability [29, 154] as well as a special case of the multivariate Tutte polynomiual [140]. A CSSG of a general graph G, is a subgraph H of G such that i) H spans G's vertex set, i.e., all of G's vertices are in H, and ii) H is connected. In other words, a CSSG of G is a subgraph of G that may lack G's edges (but not vertices), while remaining connected. For our purposes, $D(c)$ only counts a subset of all possible CSSGs of c, because we prohibit deletion of segmentations' edges.

$D(c)$ is an absolute characteristic for component c, i.e., it doesn't help us when comparing the stability of components under different bilingual segmentations. We elaborate with an example. Figure 5.5 shows two components c and c' that satisfy $D(c) = D(c') = 3$. Both components are equally difficult to be perturbed into a disconnected state, but only superficially: c contains a weakly attached subgraph to the right of vertex v. The actual structural quality of c is revealed when it is 'compared' to component $\tilde{c}$ that consists of the same source and target vertices, the same translation edges but its source vertices form exactly one chain and similarly for its target vertices; $\tilde{c}$ is essentially the 'upper bound' of c. On the other hand, the upper bound for c' is c' itself, as its source and target vertices already form exactly one chain.

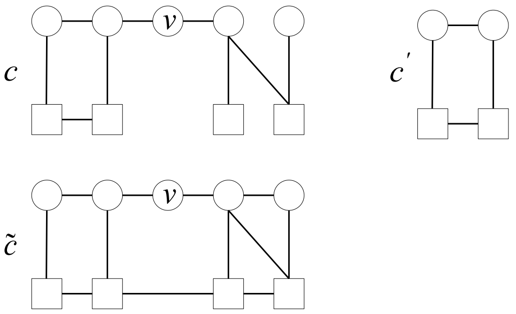


Figure 5.5: Components c and c' are such that $D(c) = D(c')$. Comparing c with $\tilde{c}$ yields c's true structural quality.

In general, the maximum value of $D(c)$, with respect to a fixed set of source and target vertices and translation edges, is attained when it consists of exactly one source chain and exactly one target chain. In such a component c, only a single translation edge is sufficient for keeping c connected. This translation edge can be *any* translation edge from a_c. Then, all combinations from $\mathcal{P}(a_c) \setminus \emptyset$ keep c connected. It follows that the desired maximum value is always $2^{|a_c|} - 1$. For any component c the structural quality measure is thus given by

$$gain(c) = \frac{D(c)}{2^{|a_c|} - 1}, \tag{5.14}$$

which takes values in $(0, 1]$. If $gain(c) = 1$, then c is fully robust with respect to its translation edges a_c. In the example of Figure 5.5, the structural quality of c and c' is thus $3/(2^5-1) = 9.7\%$ and $3/(2^2-1) = 100\%$, respectively. Intuitively, by keeping the edges of the chains fixed the quantity $gain(c)$ measures how difficult it is to perturb a component from its connected state to a disconnected state.

Finally, the measure that evaluates the structural quality of a bilingual word-aligned segmentation (σ, τ), that forms component set $C(\sigma, \tau)$, is given by

$$f(\sigma,\tau) = \left(\prod_{c \in C(\sigma,\tau)} gain(c) \right)^{\frac{1}{|C(\sigma,\tau)|}}, \tag{5.15}$$

which takes values in $(0, 1]$. We consider the geometric mean of $C(\sigma, \tau)$'s $gain$s because we want to compare different configurations with each other (different configurations lead to different sizes of components' sets). The relation $f(\sigma, \tau) > f(\sigma', \tau')$ implies that (σ, τ) is a more structurally stable bilingual segmentation than (σ', τ'). Different configurations may lead to the same structural quality. Figure 5.6 shows an example for which $f(\sigma, \tau) = f(\sigma', \tau') = 1$. Recall that our ultimate goal is to extract the bilingual natural segmentation of a word-aligned sentence pair. Apart from the structural measure of this section, a configuration is also assessed by the surface measure of Section 5.2. The top configuration of Figure 5.6 results in the sentence pair being a component as a whole, i.e., the entire source and target sentences become segments. These will most likely get a low score from the surface measure.

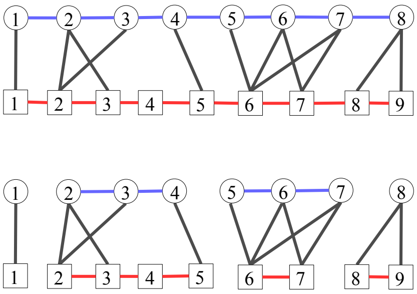


Figure 5.6: Two fully robust configurations with one (top) and four (bottom) components. Source and target vertices are shown with circles and squares respectively. Source-to-source and target-to-target edges are shown with blue and red respectively.

Observe the similarity between surface and structural measures with 'Precision' and 'Recall' respectively from Information Retrieval. Given an aligned sentence pair, a 'document' in the retrievable document collection is a component that can be found in a configuration of the sentence pair; all components from all possible configurations are represented in the document collection. The set of relevant documents is formed by those components that make up the bilingual natural segmentation of the sentence pair. Enough documents are retrieved so that all words of the sentence pair are spanned.

If all relevant documents are found in subgraphs of the retrieved documents, we then have total recall.

We conclude this section with a remark: A component with no translation edges, i.e., a source or target segment whose words are all unaligned, has a contribution of $1/0$ in (5.15). In practice we exclude such components from C.

5.3.2 Approximations

Finding the number of CSSGs of a general graph G, henceforth $\#CSSG(G)$, is a known #P-hard problem [155]. In our setting, graphs have specific formation (source and target chains connected via translation edges) and we are interested in the deletion of translation edges only; even in this case, an algorithm that computes (5.13) in polynomial time is not known. In this section, it is shown how to approximate (5.13) in linear time. Our approach leverages from the following observations:

i) Complete subcomponents As explained in the previous section, if component κ consists of exactly one source chain and exactly one target chain, then trivially $D(\kappa) = 2^{|a_\kappa|} - 1$. We henceforth refer to such component κ as *complete*. For a general component c, our aim is to investigate under which conditions c can be decomposed into complete subcomponents so that the counting process of CSSGs is trivialized. More precisely, is it possible to construct a set K of complete subcomponents with $c = \cup_{\kappa \in K} \kappa$ so that $D(c) = \prod_{\kappa \in K} (2^{|a_\kappa|} - 1)$?

ii) Spanning trees The example in Figure 5.4 shows that the search for CSSGs is graded: 0) First, inspect what happens with no translation edge deletions (returning the component itself); 1) then how many CSSGs appear if exactly *one* translation edge is deleted; 2) then how many CSSGs appear if exactly *two* translation edges are deleted, etc. This is in fact an iterative process, because each CSSG of step $i + 1)$ is given by deleting a translation edge from a CSSG in step $i)$. Unfortunately, no polynomial-time algorithm can perform this iteration.

We are content with the fact that this process starts from a set that contains component c itself and, after incrementally deleting translation edges, it arrives at the set of c's least structurally stable spanning subgraphs; this set consists of c's spanning trees. In general, for a graph $G = (V, E)$, a spanning tree T of G is a subgraph of G such that i) T spans G's vertex set, and ii) T is a tree. The associated quantity $R = |E| - |V| + 1$, which is called the *redundancy* of G, gives the maximum number of edges that can be deleted from G while G remains connected. Since the deletion of an edge from any tree results in disconnection, it follows that a) a tree has redundancy 0; b) a connected subgraph of G with R edges deleted is a tree.

Thus, for our purposes, the depth of the iterative process can be computed simply by computing the component's redundancy: By simply counting the number of edges and vertices of the component we can deduce how many steps are needed until the set of spanning trees is reached.

iii) Bijection with weighted bipartite graphs Any component c from a word-aligned bilingual segmentation has a weighted bipartite representation G_c, which can be constructed as follows: Convert each source chain into a single black-colored vertex and label it. Similarly, convert each target chain into a single white-colored vertex and label it. An edge between a black vertex and a white vertex exists if and only if their corresponding chains are connected via a translation edge. If such an edge exists, then it is assigned a weight that equals the number of translation edges between the corresponding chains.

Figure 5.7 shows an example of such a conversion. Component c consists of source chain 012 and target chains 01, 2 and 3. These are converted into single vertices u (for the source side) and α, β and γ (for the target side) respectively. Since at least one translation edge exists between source chain 012 and each of the target chains, we have that u is connected with each of α, β and γ via an edge. Also, since source chain 012 and target chain 01 are connected via three translation edges, the resulting weight of their corresponding edge in G_c is three. Similarly for the other two edges of G_c.

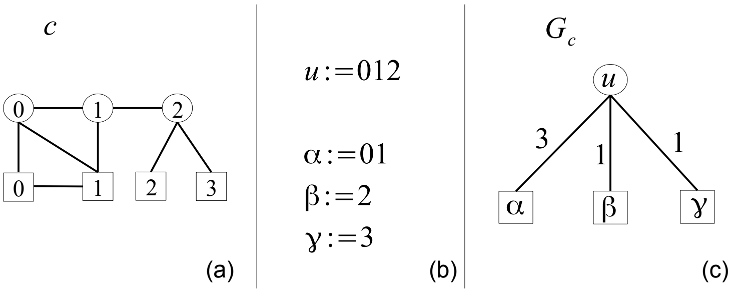


Figure 5.7: (a) Component c of a word-aligned bilingual segmentation. (b) Contracting chains of (a) into single vertices. (c) The resulting weighted bipartite graph G_c that represents component c. Edges' weights count the number of translation edges between the corresponding chains in c.

In general, weighted bipartite graph G_c is just a compact representation of component c; it does not provide further information about c. On the contrary, this representation omits certain structure, namely how translation edges actually connect complete subcomponents. This is an intentional omission as such structures will be shown to be redundant for the purpose of computing (5.13). G_c encodes all sufficient information for inferring (5.13) with minimal structure: G_c is a factorization of c into complete subcomponents and, as mentioned in observation i) above, the computation of (5.13) for complete components disregards exact source-to-target chain connectivity.

How can we then compute $D(c)$ based on G_c? It is useful to attempt to compute $\#CSSG(G_c)$ first. In the example of Figure 5.7 observe that $\#CSSG(G_c) = 1$, because G_c is a tree. This implies that G_c itself (and no subgraphs thereof) can sufficiently provide us with $D(c)$. The weight of edge $\{u, \alpha\}$ tells us that there are $2^3 - 1$ ways of deleting edges from the corresponding complete subcomponent $(012, 01)$ without violating its connectivity. Similarly for the other two edges, and we thus have

$$D(c) \;=\; (2^3 - 1)(2^1 - 1)(2^1 - 1) = 7, \tag{5.16}$$

which recovers the same value but without exhaustive search. In general, given component c, let e denote an edge in its weighted bipartite graph representation G_c ($e \in G_c$) and let w_e denote its weight. Then, we have

$$D(c) \;=\; \prod_{e \in G_c} (2^{w_e} - 1), \quad \text{if } G_c \text{ is a tree.} \tag{5.17}$$

What if G_c is not a tree? Figure 5.8 shows a component of more complex structure and its corresponding weighted bipartite representation. In this case, G_c is not a tree and we find $\#\mathit{CSSG}(G_c) = 1 + 6 + 12 = 19$ by exhaustive search: There is one way of deleting no edges from G_c (returning G_c itself); six and twelve ways of deleting one and two edges respectively while keeping G_c connected. Observe that G_c's redundancy is $6 - 5 + 1 = 2$ which verifies that we cannot go beyond two edge deletions without violating connectivity. Each CSSG of G_c provides a contribution for $D(c)$ in the same way as in the previous example. We find $D(c) = 147 + 532 + 556 = 1235$, where $147 = (2^3 - 1)^2(2^2 - 1)(2^1 - 1)^3$ for the case of deleting nothing, and similarly for all other subgraphs.

In other words, if G_c is not a tree, exhaustive search still has to be done, this time for the CSSGs of G_c. However, in general, if $R(H)$ denotes H's redundancy, then it is trivial to show that $R(c) \geq R(G_c)$. The advantage that we gain from our bijection is that the depth of the iterative process is smaller, thus making exhaustive search possible, as in the example of Figure 5.8 (for which $R(c) = 16 - 10 + 1 = 7$).

In general, let R denote the redundancy of G_c and let $\mathcal{G}_r$ denote the set of connected spanning subgraphs of G_c with r edges less than G_c. Then we have

$$D(c) \;=\; \sum_{r=0}^{R} \sum_{H \in \mathcal{G}_r} \prod_{e \in H} (2^{w_e} - 1). \tag{5.18}$$

Observe that $\mathcal{G}_0 = \{G_c\}$ and if $R = 0$, then (5.18) reduces to (5.17).

iv) Redundancy in practice Given a word-aligned sentence pair, recall that our ultimate goal is to identify its bilingual natural segmentation. As such, segments in both languages are expected to be small with components consisting of maximum two source chains and maximum two target chains. Segments of complex components as in Figure 5.8 are likely to be penalized by the surface measure. Indeed, preliminary experiments showed that 80% of the components of bilingual segmentations in the vicinity of the bilingual natural segmentation satisfied $R(G_c) = 0$. In general the larger the

$R(G_c)$, the smaller the surface score for c's segments. Thus, considering only the first couple of iterations ($r = 0, 1$) in (5.18) is a reasonable approximation.

Let $g(e) = 2^{w_e} - 1$. Then, from (5.18) we have

$$D(c) \;=\; \prod_{e \in G_c} g(e) \;+\; \sum_{r=1}^{R} \sum_{H \in \mathcal{G}_r} \prod_{e \in H} g(e). \tag{5.19}$$

If $e \in G_c \setminus H$ is read as 'edge e that is in G_c but not in H', then observe that, for any spanning subgraph H of G_c, we have

$$\prod_{e \in H} g(e) \;=\; \frac{\prod\limits_{e \in G_c} g(e)}{\prod\limits_{e \in G_c \setminus H} g(e)}. \tag{5.20}$$

Then (5.19) becomes

$$\begin{aligned} D(c) \;&=\; \prod_{e \in G_c} g(e) \;+\; \sum_{r=1}^{R} \sum_{H \in \mathcal{G}_r} \frac{\prod\limits_{e \in G_c} g(e)}{\prod\limits_{e \in G_c \setminus H} g(e)} \\ &=\; \left(\prod_{e \in G_c} g(e)\right) \left(1 \;+\; \sum_{r=1}^{R} \sum_{H \in \mathcal{G}_r} \frac{1}{\prod\limits_{e \in G_c \setminus H} g(e)}\right). \end{aligned} \tag{5.21}$$

Let

$$Q_r(c) \;=\; \sum_{H \in \mathcal{G}_r} \frac{1}{\prod\limits_{e \in G_c \setminus H} g(e)} \tag{5.22}$$

and following observation iv) above, (5.21) is approximated as

$$D(c) \;\approx\; \left(\prod_{e \in G_c} g(e)\right) (1 \;+\; Q_1(c)) \,. \tag{5.23}$$

Recall that a graph $H \in \mathcal{G}_1$ is a connected spanning subgraph of G_c that has one edge less than G_c. Thus $G_c \setminus H$ contains a single edge e, namely the one that is deleted from G_c. The fact that e *can* be deleted from G_c without violating connectivity implies that e is *not* a bridge for G_c. Let B denote the set of bridges for G_c and let $G_c \setminus B.$ denote the set of edges that are not bridges for G_c. Then we have

$$Q_1(c) \;=\; \sum_{e \in G_c \setminus B} \frac{1}{g(e)}, \tag{5.24}$$

and (5.23) becomes

$$D(c) \;\approx\; \left(\prod_{e \in G_c} g(e)\right)\left(1 \;+\; \sum_{e \in G_c \setminus B} \frac{1}{g(e)}\right). \tag{5.25}$$

For a general graph $G = (V, E)$, finding the bridges of G requires $O(|V| + |E|)$ time [146]. Thus, the computation of (5.25) requires linear time in the number of vertices and edges in G_c. During exploratory data analysis we found that $D(c) = \prod_{e \in G_c} g(e)$ is an equally good approximation to (5.25). Hence, (5.14) becomes

$$gain(c) \;=\; \frac{\prod\limits_{e \in G_c} (2^{w_e} \;-\; 1)}{2^{\sum_{e \in G_c} w_e} \;-\; 1}\;. \tag{5.26}$$

5.4 Extracting bilingual segmentations with the Cross-Entropy method

Equipped with the measures of Sections 5.2 and 5.3 we turn to extracting an N-best list of bilingual segmentations for a given sentence pair. Let $F[X, Y]$ denote the harmonic mean of X and Y. Given configuration (σ, τ), let $g(\sigma, \tau)$ and $f(\sigma, \tau)$ denote its surface and structural quality score respectively. The latter is given by (5.15) and the former by $g(\sigma, \tau) = F[g(\sigma), g(\tau)]$, where $g(\omega)$ can be generated by method `VLM` or `REF` of Section 5.2. The measure that evaluates the extent to which configuration (σ, τ) jointly meets Conditions 1 and 2 of Section 5.1 is given by $F[g(\sigma, \tau), f(\sigma, \tau)]$. In this section, we explain how to find

$$(\sigma^*, \tau^*) \;=\; \arg\max_{\sigma, \tau}\; F[g(\sigma, \tau), f(\sigma, \tau)], \tag{5.27}$$

where the search is over all continuous configurations, which is exponential in the total number of words of the sentence pair. We propose a new approach for this task, by noting a direct connection with the combinatorial problems that can be solved efficiently and effectively with the Cross-Entropy (CE) method [128].

The CE method is an iterative self-tuning sampling method that has applications in various combinatorial and continuous global optimization problems as well as in rare event detection [129]. A detailed account on the CE method can be found in Appendix B. Here, we simply describe its application to our problem.

A segmentation of a sentence has a bit string representation in the following way: If two consecutive words in the sentence belong to the same segment in the segmentation, then this pair of words is encoded by ‘1’, otherwise by ‘0’. The bit string representation of a bilingual segmentation of a sentence pair, is given by concatenating the bit string representations of the two sentences. Such a representation is bijective and, thus, for the rest of this section, we do not distinguish between a bilingual segmentation and its bit string representation. As in Section 4.4, the algorithm of the CE method is a repeated application of (a) sampling bit strings from a parametrized probability mass function, (b) scoring them and keeping only a small high-performing subsample, and (c) updating the parameters of the probability mass function based on that subsample only.

Samples are bit strings of length $l = n + m - 2$, where n and m are the number of words in the source and target sentence respectively. The first $n - 1$ bits correspond to the source segmentation and the rest to the target segmentation. As mentioned above, the surface quality score of such a bit string is given by the harmonic mean of its source and target surface quality scores. However, as mentioned in Section 5.2, surface quality score $g(\omega)$ of segmentation ω is a real number. At each iteration of the algorithm $g(\omega)$ is converted into a number in $(0, 1]$ via Min-Max normalization.

No prior knowledge is assumed on the quality of bit strings, so that they are all equally likely: Each position of a randomly chosen bit string can be either a '0' or a '1' with probability $1/2$. The aim is to tune these position probabilities towards the best bit string, with respect to scoring function F. A bit string in a sample is labeled by integer i; it is denoted by x_i and its jth bit is denoted by x_{ij}. Set

- $M = 20l$, the sample size;
- $\rho = \lceil 1\% M \rceil$ and $\Lambda = 5\rho$, numbers indicating how many best-performing bit strings of a sample are selected;
- $\alpha = 0.7$, the smoothing parameter;
- indicator function $I_A = 1$, if event A occurs and $I_A = 0$, otherwise.

The algorithm for finding (5.27) is as follows:

1. Set $(\theta_1^0, ..., \theta_l^0) = (1/2, ..., 1/2)$ and $t = 1$.
2. 2.1. Generate sample $x_1, ..., x_M$ of bit strings, each of length l, such that $x_{ij} \sim$ Bernoulli(θ_j^{t-1}), for all $i = 1, ..., M$ and $j = 1, ..., l$.

   2.2. Compute surface scores $g(x_1), ..., g(x_M)$ of x_1,...,x_M.

   2.3. Order them descendingly as $g(x_{\pi(1)}) \geq ... \geq g(x_{\pi(M)})$, where π is the associated permutation of $\{1, ..., M\}$.

   2.4. Compute harmonic mean scores $F(x_{\pi(1)}), ..., F(x_{\pi(\Lambda)})$ of $x_{\pi(1)}$,...,$x_{\pi(\Lambda)}$.

   2.5. Order them descendingly as $F(x_{\phi(1)}) \geq ... \geq F(x_{\phi(\Lambda)})$, where ϕ is the associated permutation of $\{1, ..., \Lambda\}$.
3. Compute performance threshold $\gamma^t = F(x_{\phi(\rho)})$.

4. Compute bit probabilities

$$\theta_j^t = \frac{\sum_{i=1}^{M} I_{\{F(x_i) \geq \gamma^t\}} \; x_{ij}}{\sum_{i=1}^{M} I_{\{F(x_i) \geq \gamma^t\}}}, \qquad j = 1, ..., l. \tag{5.28}$$

5. Smooth bit probabilities

$$\theta_j^t := \alpha \theta_j^t + (1 - \alpha) \theta_j^{t-1}, \qquad j = 1, ..., l. \tag{5.29}$$

6. If $t \geq 5$ and $\gamma^t = \gamma^{t-1} = ... = \gamma^{t-5}$, then stop. Else $t := t + 1$ and go to Step 2.

In Step 2.2, why do we not simply compute harmonic mean scores $\{F(x_i)\}_{i=1}^{M}$? The computation of F requires the computation of structural score f, which in its turn requires the identification of connected components of a configuration/bit string. In practice, we found that the latter is computationally expensive when performed for all configurations of the sample. We chose to firstly assess bit strings of the whole sample simply by their surface quality, which is computationally cheap, and identify a g-high performing subsample; this subsample, which consists of Λ bit strings, is then resorted according to F. This is a valid approximation as bit strings with low surface score are bound not to be included in the final F-highest performing bit strings. We thus save computational time by not identifying connected components of poor configurations. The subsample size of $\Lambda = 5\rho$, i.e., five times larger than the final optimal subset of an iteration, was found to work well in practice.

The values for parameters M, ρ and α reported here are in line with the ones suggested in the literature [129] for combinatorial problems such as this one. After the execution of the algorithm, the updated vector of position probabilities converges to sequence of '0's and '1's, which corresponds to the best bilingual segmentation under F. Finally, N-best lists are trivially generated, simply by collecting the top-N performing accumulated samples of a maximization process.

5.5 Experiments

Given a sentence pair with known and fixed word alignments, the result of the algorithm described in Section 5.4 is an N-best list of bilingual segmentations of such a pair. The objective function provides a balance between compositional expressive power of segments in both languages and source-target segments' synchronization via word alignments. Each (continuous) component of such a bilingual segmentation leads to the extraction of a phrase pair that becomes a translation rule for SMT.

As mentioned in Section 5.1, a translation rule of phrase-based SMT is constructed from a component or from the union of components of an unsegmented word-aligned sentence pair. For each sentence pair, all possible (continuous) components and (continuous) unions of components give rise to the extracted (continuous) phrase pairs. In this section we investigate the impact on SMT models and translation quality, when extracting phrase pairs (from the N-best lists) that correspond to components *only*. A reduction in phrase-table size is guaranteed because we are essentially extracting only a subset of all possible continuous phrase pairs. The challenge is to verify whether this subset can provide a sufficient translation model.

Four variants of the same experiment are carried out: The surface measure is powered by either method `VLM` or method `REF` and the structural measure is either included normally or absent. If (S, T) denotes a word-aligned sentence pair then let $C_{S,T}$ denote the set of components of (S, T) and let $C_{S,T}(\sigma, \tau)$ denote the set of components of a bilingual segmentation of (S, T). Also, for any word-aligned sentence pair, let (σ_0, τ_0) denote the exceptional segmentation for which each segment is a unigram for both sentences, i.e, the case where (S, T) is unsegmented and forms component set $C_{S,T} \equiv C_{S,T}(\sigma_0, \tau_0)$. For a given word-aligned sentence pair, let $\{(\sigma_i, \tau_i)\}_{i=1}^{N}$ denote an N-best list that is generated by (5.27). By considering all word-aligned sentence pairs in the training data, the sets of translation rules that form the phrase-tables are given by

$$P_N[m, f] \;=\; \bigcup_{S,T} \bigcup_{i=0}^{N} \{\, p \;:\; p \in C_{S,T}(\sigma_i, \tau_i) \,\}, \tag{5.30}$$

where $m = $ `VLM`, `REF` shows which method is used for the surface measure and f is either given by (5.27) or $f = 1$ that indicates its complete absence. The set of translation rules that forms the baseline phrase-table is given by

$$P \;=\; \bigcup_{S,T} \{\, p \;:\; p \in \mathcal{P}(C_{S,T}) \,\}. \tag{5.31}$$

Both the baseline and our system are standard phrase-based MT systems. Bidirectional word alignments are generated with GIZA++ [111] and 'grow-diag-final-and' heuristic. These are used to construct a phrase-table with bidirectional phrase probabilities, lexical weights and a reordering model with monotone, swap and discontinuous orientations, conditioned on both the previous and the next phrase. 4-gram interpolated language models with Kneser-Ney smoothing are built with SRILM [144]. A distortion limit of 6 and a phrase-penalty are also used. All model parameters are tuned with MERT [110]. Decoding during tuning and testing is done with Moses [85]. Since our system only affects which phrases are extracted, lexical weights and reordering orientations are the same for both systems.

Datasets are from the WMT'13 translation task [14]: Translation and reordering models are trained on Czech–English and German–English corpora (Table 5.1). Language models and surface measures `VLM` and `REF` are trained on 35.3M Czech, 50.0M German and 94.5M English sentences from the provided monolingual data. Tuning is done on newstest2010 and performance is evaluated on newstest2008, newstest2009, newstest2011 and newstest2012 with BLEU [115].

The size of an N-best list varies according to the total number of words in the sentence pair, say w. For the purposes of phrase extraction in SMT we would ideally require all local maxima to be part of an N-best list. This would guarantee the

	Cz–En	De–En
Europarl (v7)	642,505	1,889,791
News Commentary (v8)	139,679	177,079
Total	782,184	2,066,870

Table 5.1: Number of filtered parallel sentences for Czech–English and German–English.

Method	Czech→English				English→Czech				CZ–EN
	'08	'09	'11	'12	'08	'09	'11	'12	PT size
P	19.6	20.6	22.6	20.6	14.8	15.6	16.6	14.9	44.6M (100%)
$P_N[\texttt{VLM}, 1]$	19.2	20.1	21.8	20.0	14.2	14.7	16.3	14.3	5.8M (13.0%)
$P_N[\texttt{REF}, 1]$	19.3	20.2	22.3	20.1	14.4	14.9	16.2	14.5	4.7M (10.5%)
$P_N[\texttt{VLM}, f]$	19.5	20.4	22.2	20.3	14.4	15.1	16.5	14.6	7.5M (16.8%)
$P_N[\texttt{REF}, f]$	19.6	20.5	22.6	20.7	14.6	15.4	16.8	14.7	4.6M (10.4%)

Table 5.2: BLEU scores and phrase-table (PT) sizes for Czech–English. Phrase-table of method P, the baseline, is constructed from all consistent phrase pairs. Phrase-tables of methods $P_N[m, 1]$, for $m = \texttt{VLM}, \texttt{REF}$, are constructed from consistent phrase pairs that are extracted from components of the top-N word-aligned bilingual segmentations and (σ_0, τ_0) of each sentence pair, without taking into account the structural measure. Similarly for $P_N[m, f]$, for $m = \texttt{VLM}, \texttt{REF}$, but with the structural measure included.

extraction of all high quality phrase pairs, with (empirically) desired variations, while keeping N small. Since the CE method performs global optimization, the resulting members of an N-best list are in the vicinity of the global maximum. Consequently, we cannot guarantee the inclusion of local maxima. We set $N = \lceil 30\% w \rceil$ so that at least some variation from the global maximum is included, but this value is not large enough to contaminate the lists with noisy bilingual segmentations. The resulting lists have 22 bilingual segmentations on average for both language pairs. Figure 5.9 shows typical German–English best performing bilingual segmentations for which REF was used for the surface measure and the structural measure was included.

BLEU scores are reported in Tables 5.2 and 5.3 for Czech–English and German–English, respectively. Methods P, $P_N[m, 1]$ and $P_N[m, f]$, for $m = \texttt{VLM}, \texttt{REF}$, are the ones described above. Phrase-table sizes are reduced in all cases as expected. The best performing method that achieves phrase-table reduction is $P_N[\texttt{REF}, f]$ and is comparable to P, the baseline. In fact, the translation quality of this method suggests that the set of components of the bilingual natural segmentation of a word-aligned sentence pair is indeed the one that contributes to the effective set of translation rules. On the other hand, if the surface measure is generated by VLM translation quality is consistently worse. It is also evident that the inclusion of the structural measure is

Method	German→English				English→German				DE–EN PT size
	’08	’09	’11	’12	’08	’09	’11	’12	
P	21.4	20.8	21.3	22.1	15.1	15.1	16.0	16.5	102.3M (100%)
$P_N[\texttt{VLM}, 1]$	21.1	20.3	20.9	21.4	14.9	14.7	15.3	16.1	11.4M (11.1%)
$P_N[\texttt{REF}, 1]$	21.3	20.6	21.0	21.9	15.2	14.9	15.5	16.1	9.6M (9.4%)
$P_N[\texttt{VLM}, f]$	21.4	20.6	21.3	21.7	15.3	14.9	15.6	16.3	16.8M (16.4%)
$P_N[\texttt{REF}, f]$	21.5	20.8	21.5	22.0	15.4	15.2	15.7	16.2	9.9M (9.7%)

Table 5.3: Similar to Table 5.2, but for German–English.

imperative, regardless of how the surface measure is chosen.

We show that translation rules emerging from unsegmented word-aligned sentence pairs are essential in forming the effective set of translation rules. Let the set

$$P^*_N[\texttt{REF}, f] \;=\; \bigcup_{S,T} \bigcup_{i=1}^{N} \{\, p \;:\; p \in C_{S,T}(\sigma_i, \tau_i) \,\}, \tag{5.32}$$

denote all translation rules that are extracted from N-best lists only, using method REF for the surface measure and with the structural measure included. Tables 5.4 and 5.5 show the comparison with $P_N[\texttt{REF}, f]$. It is evident that excluding translation rules that emerge from unsegmented word-aligned sentence pairs harms translation quality. In these tables, our best performing method, $P_N[\texttt{REF}, f]$, is also compared to $P_{\alpha+\epsilon}$, which is the significance pruning technique of Johnson et al. [74] with parameter $\alpha+\epsilon$. Significance pruning is entirely based on statistics of phrase pairs and as such it lacks linguistic motivation. In particular, it tests whether a source phrase and a target phrase co-occur more frequently in a bilingual corpus than they should just by chance. To this end, Fisher’s exact test is employed and $\alpha + \epsilon$ is the pruning threshold for the associated p-values. Results indicate that translation quality of all three methods P, $P_{\alpha+\epsilon}$ and $P_N[\texttt{REF}, f]$ is comparable.

5.6 Related work

The sizes of the resulting phrase-tables together with the type of phrase pairs that are extracted lead to applications involving discontinuous phrase pairs. In [54] there was evidence that discontinuous phrase pairs that are extracted from discontinuous components of word-aligned sentence pairs can improve translation quality. As the number of such components is much bigger than the continuous ones, Gimpel and Smith [56] propose a Bayesian nonparametric model for finding the most probable discontinuous phrase pairs. This can also be done from the N-best lists that are generated in Section 5.4, and it would be interesting to see the effect of such phrase pairs in our existing models.

Method	Czech→English				English→Czech				CZ–EN PT size
	’08	’09	’11	’12	’08	’09	’11	’12	
P	19.6	20.6	22.6	20.6	14.8	15.6	16.6	14.9	44.6M (100%)
$P_{\alpha+\epsilon}$	19.6	20.4	22.4	20.5	14.5	15.5	16.8	14.8	3.7M (8.3%)
$P^*_N[\texttt{REF}, f]$	19.7	20.4	22.4	20.3	14.4	15.2	16.3	14.3	4.4M (9.8%)
$P_N[\texttt{REF}, f]$	19.6	20.5	22.6	20.7	14.6	15.4	16.8	14.7	4.6M (10.4%)

Table 5.4: BLEU scores and phrase-table (PT) sizes for Czech–English. Phrase-table of method P, the baseline, is constructed from all consistent phrase pairs. Phrase-table of method $P_{\alpha+\epsilon}$ is constructed from pruning P using the method of Johnson et al. (2007) with parameter $\alpha + \epsilon$. Phrase-table of method $P^*_N[\texttt{REF}, f]$, is constructed from consistent phrase pairs that are extracted from components of the top-N word-aligned bilingual segmentations of each sentence pair, for which the surface measure is generated by method REF and by taking into account the structural measure. Similarly for $P_N[\texttt{REF}, f]$, but with translation rules from unsegmented word-aligned sentence pairs included.

Method	German→English				English→German				DE–EN PT size
	’08	’09	’11	’12	’08	’09	’11	’12	
P	21.4	20.8	21.3	22.1	15.1	15.1	16.0	16.5	102.3M (100%)
$P_{\alpha+\epsilon}$	21.4	20.8	21.6	22.0	15.3	15.5	16.1	16.6	9.9M (9.7%)
$P^*_N[\texttt{REF}, f]$	21.3	20.6	21.3	21.8	15.0	15.0	15.6	16.0	9.4M (9.2%)
$P_N[\texttt{REF}, f]$	21.5	20.8	21.5	22.0	15.4	15.2	15.7	16.2	9.9M (9.7%)

Table 5.5: Similar to Table 5.4, but for German–English.

As mentioned on many occasions in this thesis, the process of obtaining the most likely alignment for a sentence pair forms a natural partition for the pair and consequently a basic bilingual segmentation for the pair. In [130] this process was modified in order to obtain an alignment that is (statistically) tailored to the source sentence of the pair, thus constructing source-driven bilingual segmentations (SDBS). The motivation lies in constructing translation rules that would be suitable for decoding: The process of decoding requires the segmentation of an unseen source-type sentence; the source phrases of SDBS could potentially be good matches for these segments and the target phrases of SDBS would form a potentially more accurate translation. Although phrase-tables smaller than the baseline were observed, the desired results were not achieved. Motivated by this work, we also attempted to construct SDBS by setting $g(\sigma, \tau) = g(\sigma)$ in (5.27), but indifferent overall results were obtained (phrase-table sizes and translation quality were similar to the ones in Section 5.5).

5.7 Conclusions

The purpose of this chapter has been twofold: First, to generalize the concept of natural segmentation that was introduced Chapter 4 into a bilingual setting:

RQ5 *Given a word-aligned sentence pair, identify what conditions a bilingual segmentation of the pair should satisfy in order to form a bilingual natural segmentation. How can one define the bilingual segmentation that satisfies those conditions optimally?*

This was achieved by stating two conditions that a natural bilingual segmentation should satisfy.

1. Segments in both source and target language sentences abide to natural segmentation.

2. Source and target language segments are synchronized with each other via word alignments.

Condition 1 was discussed in Chapter 4: A natural segmentation of a sentence in any language is a segmentation whose segments provide succinctly their grammatical/syntactic/semantic role in the sentence and they do so minimally. In the bilingual setting, both sentences in a sentence pair are expected to be in line with this type of segmentation. Condition 2, which was the focus of this chapter, has been identified as the necessary condition that permits the said generalization.

It refers to purely structural properties of a degenerate graph representation of a given word-aligned bilingual segmentation. The lexical meaning of words is overlooked and they are simply treated as labelled vertices. The focus has been on the set of components of such a representation. The reason lies in the core nature of these objects in SMT: Translation rules that form the phrase-table are extracted from components and unions of components of word-aligned sentence pairs. Inspection of the effective subset of all translation rules indicated the potential connection with bilingual segments that abide to natural segmentations.

To this end, our aim was to identify what constitutes a robust component, thus making it a candidate for a member of a bilingual natural segmentation. Surely, Condition 1 already captures certain robustness criteria, but they operate independently in the two sentences of a pair. By focusing on the structural properties of components we assessed whether source and target segments are synchronized in a word-aligned bilingual segmentation. Based on the notion of connecting spanning subgraph (CSSG) from graph theory, we introduced a measure that assesses structural robustness of components. It counts a particular type of CSSGs of a component, namely the connected spanning subgraphs of the component from which only translation edges are allowed to be deleted. The appropriately normalized value of this quantity essentially tells how difficult it is to perturb the component from its connected state into a disconnected state.

Conditions 1 and 2 give rise to surface and structural robustness quantitative criteria respectively. The former can be viewed as ‘Precision’ and the latter as ‘Recall’. Thus, a bilingual natural segmentation was defined as a bilingual segmentation that provides a balance between Precision and Recall, i.e., the harmonic mean of surface and structural measures. The Cross-Entropy method that was discussed in Chapter 4 was trivially adapted to the problem of efficiently identifying the bilingual segmentation that maximizes such a harmonic mean.

The second task of this chapter was to assess the impact of bilingual natural segmentations to the translation model of SMT:

RQ6 *What is the effect of bilingual natural segmentations on SMT?*

Phrase-tables were formed by extracting translation rules from components only (and not unions thereof) of configurations in the vicinity of the bilingual natural segmentations. Translation quality was shown to be comparable with the case where the phrase-table is formed from extracting all valid translation rules, while being dramatically smaller in size. We concluded that translation rules of use to SMT have a specific structure: Basic phrase pairs, or building blocks of bilingualism (the equivalent of words/tokens in monolingualism), together with larger phrase pairs that have a stand-alone or an almost stand-alone function in sentence pairs, but are of minimal or almost minimal size.

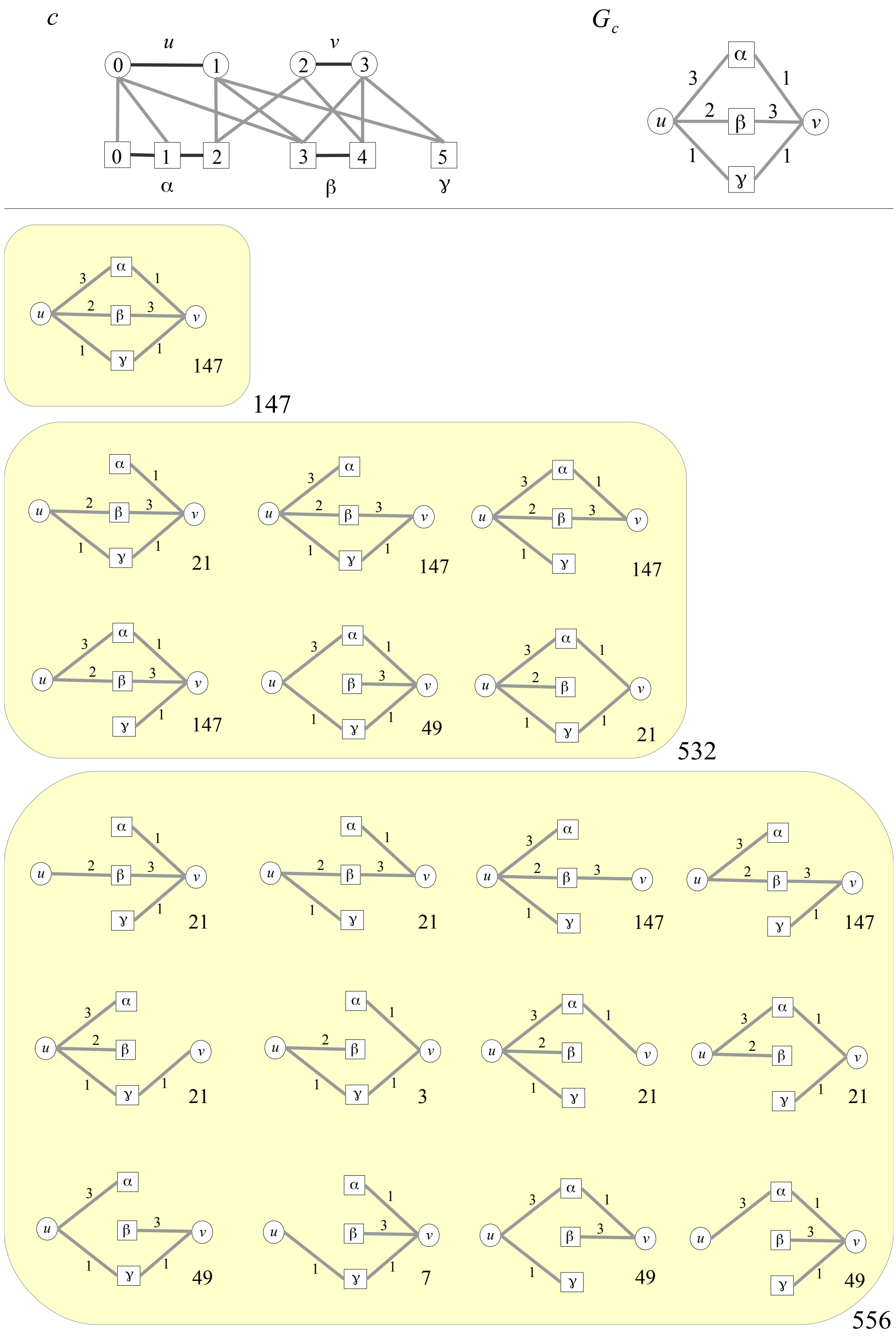


Figure 5.8: Component c, its weighted bipartite representation G_c and all CSSGs of the latter. Each number to the bottom right of a CSSG contributes to $D(c)$.

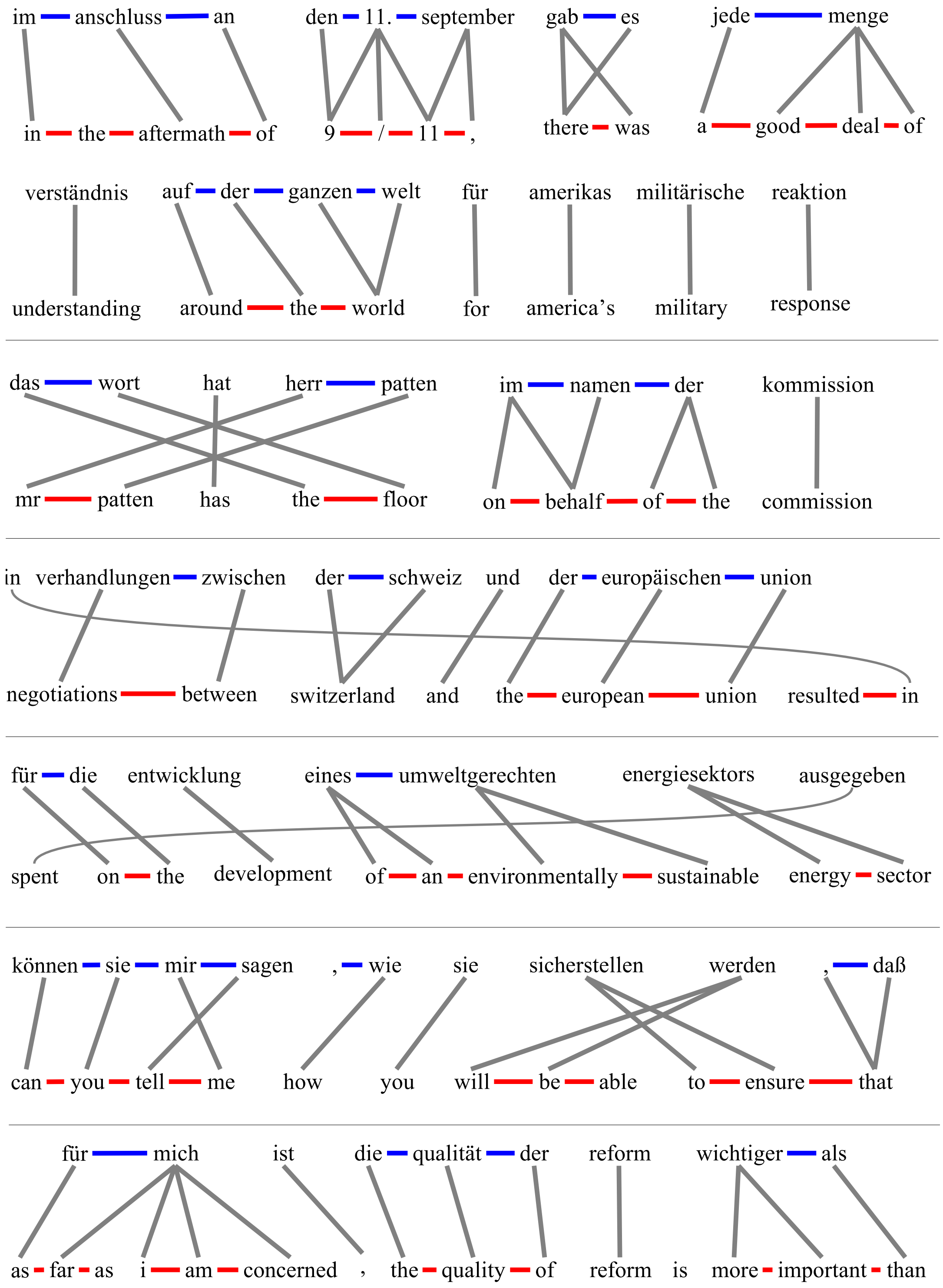

im anschluss an
in the aftermath of
den 11. september
9 / 11 ,
gab es
there was
jede menge
a good deal of
verständnis
understanding
auf der ganzen welt
around the world
für
for
amerikas
america's
militärische
military
reaktion
response
das wort hat herr patten
mr patten has the floor
im namen der
on behalf of the
kommission
commission
in verhandlungen zwischen der schweiz und der europäischen union
negotiations between switzerland and the european union resulted in
für die entwicklung eines umweltgerechten energiesektors ausgegeben
spent on the development of an environmentally sustainable energy sector
können sie mir sagen , wie sie sicherstellen werden , daß
can you tell me how you will be able to ensure that
für mich ist die qualität der reform wichtiger als
as far as i am concerned , the quality of reform is more important than

Figure 5.9: Typical fragments from best performing German–English segmentations.

Chapter 6

Paraphrases from Bilingual Aligned Corpora

The aim of this chapter is to extend existing graph-based methods that extract paraphrases from aligned bilingual corpora, as stated in **RQ7**. The graph under consideration is the graph representation of the phrase-table that results from aligned bilingual corpora: Source and target phrases are viewed as vertices and phrase-table entries represent edges. In line with previous work, we exploit the connectivity of such a graph in order to find paraphrases for a given source phrase that exists in the graph and similarly for a target phrase. Our method builds on forming a collection of sub-phrase-tables which is mainly constructed from connected components of the phrase-table's graph representation. The connectivity of each sub-phrase-table is further exploited for the purpose of creating separate source phrase and target phrase clusters. Each cluster inherits a new weighted graph structure from its corresponding sub-phrase-table. A random walk distance measure, namely the *commute time* between two vertices, is then employed for finding the degree of similarity between any two phrases in a cluster. An important distinction of this method from previous work is that a pair of phrases may appear in multiple clusters across multiple sub-phrase-tables. This allows us to compute their similarity on average, thus significantly reducing the effects of noise. Furthermore, using a novel technique, these average distances are converted into *counts*. These counts are interpreted as artificial co-occurrence counts between a pair of source phrases in hypothetical source language-to-paraphrased source language aligned corpora.

6.1 Introduction

In statistical machine translation the process of translating an unseen source language sentence into a target language sentence is known as decoding. It requires the existence of models that consist of finite and fixed source-to-target language rules. As described in Chapters 1 and 2, these models are learned from aligned parallel corpora. Such rules are source-to-target phrase translation rules, reordering rules and possibly syntax-based rules. They emerge directly from the aligned parallel corpus (with associated statistics) and are restricted to it: A valid source-to-target language rule that has not been observed in the parallel corpus, cannot be part of the models. Furthermore, the construction of the models does not suggest a direct way of generating an unseen valid rule.

During decoding, the source sentence is segmented in a way so that its segments can identify potential translations via the set of rules. The larger the source segment, the more reliable its potential translation. This is because, generally, larger source segments capture enough context to deem them as univocal with well-defined local reordering structure with the target side. Thus, typically, their translations are few and are exact paraphrases of each other. However, the likelihood of finding such an entry in the set of rules becomes smaller as the length of the source segment becomes larger. The inability to find appropriate translation candidates, or a low-quality preliminary translated sentence leads to gradual shortening of source segments. Clearly, for shorter segments, the likelihood of identifying potential translations becomes larger. However, if the segmentation contains mostly shorter segments, then the risk of forming a low-quality translation also increases.

A way to overcome this problem is by creating new rules from the set of existing ones. In this chapter, we focus on the first step prior to augmenting the phrase-table with new translation rules. This involves the identification of paraphrases for a given source phrase. Paraphrase extraction has emerged as an important problem in NLP: There exists an abundance of methods for extracting paraphrases from monolingual, comparable and bilingual corpora [3, 94]. We focus on the latter and specifically on the phrase-table that is extracted from a bitext during the training stage of SMT. Bannard and Callison-Burch [6] introduced the *pivoting* approach, which relies on a 2-step transition from a phrase, via its translations, to a paraphrase candidate. By incorporating the syntactic structure of phrases [23], the quality of the paraphrases extracted with pivoting can be improved: Kok and Brockett [87], henceforth KB, used a random walk framework to determine the similarity between phrases, which was shown to outperform pivoting with syntactic information, when multiple phrase-tables are used. In SMT, extracted paraphrases with associated pivot-based [21, 112] and cluster-based [89] probabilities have been found to improve the quality of translation. Pivoting has also been employed in the extraction of syntactic paraphrases, which are a mixture of phrases and non-terminals [55, 167]. In this chapter the following research question is addressed:

RQ7 *How should one extend the work of Kok and Brockett [87] in order to identify less noisy pairs of paraphrases and to develop a method that constructs artificial co-occurrence counts for these pairs?*

We develop a method for extracting paraphrases from a bitext for both the source and target languages. Emphasis is placed i) On the quality of the phrase-paraphrase probabilities, i.e., the conditional probabilities $p(u|v)$ and $p(v|u)$, where u and v are phrases, and ii) On providing a stepping stone for extracting syntactic paraphrases with equally reliable probabilities. In line with previous work [6, 87], our method depends on the connectivity of the phrase-table, but the resulting construction treats each side separately, which can potentially benefit from additional monolingual data.

The initial problem in harvesting paraphrases from a phrase-table is the identification of the search space. Previous work has relied on breadth first search from the query phrase with a depth of two (pivoting) and six (KB). The former can be too restrictive and the latter can lead to excessive noise contamination when taking shallow syntactic information features into account. Instead, we choose to cluster the phrase-table into separate source and target clusters and in order to make this task computationally feasible, we decompose the phrase-table into sub-phrase-tables. We propose a novel heuristic algorithm for the decomposition of the phrase-table, and use a well-established co-clustering algorithm for clustering each sub-phrase-table.

The underlying connectivity of the source and target clusters gives rise to a natural graph representation for each cluster. The vertices of the graphs consist of phrases and features with a dual smoothing/syntactic-information-carrier role. The latter allow for (a) the redistribution of the mass for phrases with no appropriate paraphrases and (b) the extraction of syntactic paraphrases. The proximity among vertices of a graph is measured by means of a random walk distance measure, the *commute time* [2]. This measure is known to perform well in identifying similar words on the graph of WordNet [123] and a related measure, the *hitting time* is known to perform well in harvesting paraphrases on a graph constructed from multiple phrase-tables (KB).

Generally in NLP, power-law distributions are typically encountered in the collection of counts during the training stage. Commute times between phrases on the said graph are converted into artificial co-occurrence counts with a novel technique. Although they need not be integers, the main challenge is the type of the underlying distributions; it should ideally emulate the resulting count distributions from the phrase extraction stage of a monolingual parallel corpus [46]. These counts give rise to the desired probability distributions by means of relative frequencies.

6.2 Constructing sub-phrase-tables

For the decomposition of the phrase-table into sub-phrase-tables it is convenient to view the phrase-table as an undirected, unweighted graph P with the vertex set being the source and target phrases and the edge set being the phrase-table entries. For the rest of this section, we do not distinguish between source and target phrases, i.e., both types are treated equally as vertices of P. When referring to the size of a graph, we mean the number of vertices it contains.

A trivial initial decomposition of P is achieved by identifying all its *connected components* (components for brevity), i.e., the mutually disjoint connected subgraphs, $\{P_0, P_1, ..., P_n\}$. As we shall see in Section 6.6.1, the largest component, say P_0, can be of significant size. We call P_0 *giant* and it needs to be further decomposed.

This is done by identifying the set of cut-vertices of the component. A *cut-vertex* (or *articulation point*) of a component is a vertex with the following property: If this vertex is removed (with its boundary edges) from the component, then the component becomes disconnected. Cut-vertices of high connectivity degree are removed from the giant component. For the remaining vertices of the giant component, new components are identified and we proceed iteratively, while keeping track of the cut-vertices that are removed in each iteration, until the size of the largest component is less than a certain threshold θ.

At each iteration, when removing cut-vertices from a giant component, the resulting collection of components may include graphs consisting of a single vertex. We refer to such vertices as *residues*. They are excluded from the resulting collection and are considered for separate treatment, as explained later in this section.

The cut-vertices need to be inserted appropriately back to the components: Starting from the last iteration step, the respective cut-vertices are added to all the components of P_0 that they used to 'glue' together; this process is performed iteratively, until there are no more cut-vertices to add. By 'addition' of a cut-vertex to a component, we mean the re-establishment of edges between the former and other vertices of the latter. The result is a collection of components whose total number of unique vertices is less than the number of vertices of the initial giant component P_0.

These remaining vertices are the residues. We then construct the graph R that consists of the residues together with *all* their translations (even those that are included in components of the above collection) and then identify its components $\{R_0, ..., R_m\}$. It turns out that the largest component, say R_0, is giant and we repeat the decomposition process that was performed on P_0. This results in a new collection of components as well as new residues: As we shall see in Section 6.6.1 the components need to be pruned and the residues give rise to a new graph R' that is constructed in the same way as R. We proceed iteratively until the number of residues stops changing. For each remaining residue u, its translations are identified, and for each translation v we identify the largest component of which v is a member and add u to that component.

The final result is a collection $\mathcal{C} = \mathcal{D} \cup \mathcal{F}$, where $\mathcal{D}$ is the collection of components emerging from the entire iterative decomposition of P_0 and R, and $\mathcal{F} = \{P_1, ..., P_n\}$. Figure 6.1 shows an example of a decomposition of a connected graph G_0; it is assumed for simplicity that only one cut-vertex is removed at each iteration and ties are resolved arbitrarily. In Figure 6.2 the residue graph is constructed and its two components are identified. The iterative insertion of the cut vertices is also depicted. The resulting two components together with those from R form the collection $\mathcal{D}$ for G_0.

The addition of cut-vertices into multiple components, as well as the construction method of the residue-based graph R, can yield the occurrences of a vertex in multiple components in $\mathcal{D}$. We exploit this property in two ways:

(a) In order to mitigate the risk of excessive decomposition (which implies greater risk of good paraphrases being in different components), as well as to reduce the size of $\mathcal{D}$, a conservative merging algorithm of components is employed. Suppose that the

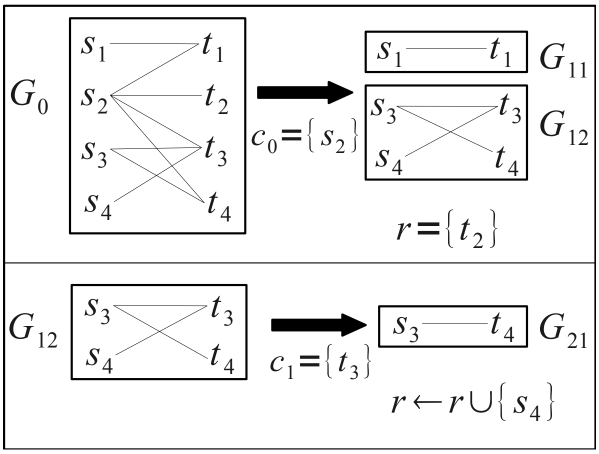


Figure 6.1: The decomposition of G_0 with vertices s_k and t_l: The cut-vertex of the ith iteration is denoted by c_i, and r collects the residues after each iteration. The task is completed in Figure 6.2.

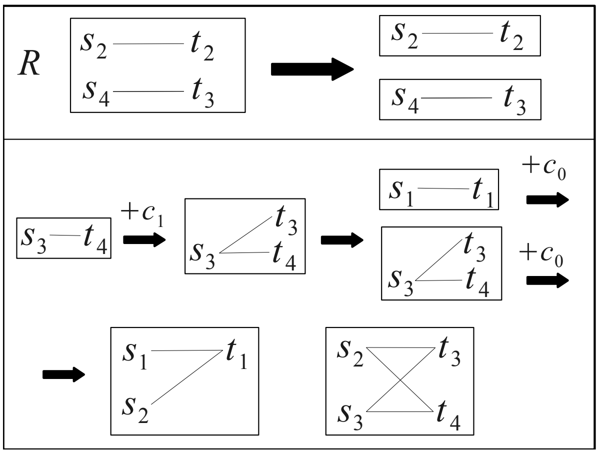


Figure 6.2: Top: Residue graph with its components (no further decomposition is required). Bottom: Adding cut-vertices back to their components.

elements of $\mathcal{D}$ are ranked according to size in ascending order as

$$\mathcal{D} = \{D_1, ..., D_k, D_{k+1}, ..., D_{|\mathcal{D}|}\}, \quad \text{where } |D_i| \leq \delta, \tag{6.1}$$

for $i = 1, ..., k$ and some threshold δ. Each component D_i with $i \in \{1, ..., k\}$ is examined as follows: For each vertex of D_i the number of its occurrences in $\mathcal{D}$ is inspected. This is done in order to identify an appropriate vertex b to act as a bridge between D_i and other components of which b is a member. Note that translations of a vertex b with smaller number of occurrences in $\mathcal{D}$ are less likely to capture their full spectrum of paraphrases. We thus choose a vertex b from D_i with the smallest number of occurrences in $\mathcal{D}$, resolving ties arbitrarily, and proceed with merging D_i with the largest component, say D_j with $j \in \{1, ..., |\mathcal{D}| - 1\}$, of which b is also a member. The resulting merged component $D_{j'}$ contains all vertices and edges of D_i and D_j and new edges, which are formed according to the rule: if u is a vertex of D_i and v is a vertex of D_j and $\{u, v\}$ is a phrase-table entry, then (u, v) is an edge in $D_{j'}$. As long as no connected component has identified D_i as the component with which it should be merged, then D_i is deleted from the collection $\mathcal{D}$.

(b) In Information Retrieval, the inverse document frequency (idf) is a statistical weight used for measuring the importance of a term in a collection of text documents [126, 141]. We define an idf-inspired measure for each phrase pair (x, x') of the same type (source or target) as

$$idf(x, x') = \frac{1}{\log |\mathcal{D}|} \log \left(\frac{2c(x, x')|\mathcal{D}|}{c(x) + c(x')} \right), \tag{6.2}$$

where $c(x, x')$ is the number of components in which the phrases x and x' co-occur, and equivalently for $c(\cdot)$. The purpose of this measure is for pruning paraphrase candidates and its use is explained in Section 6.4. Note that here $idf(x, x') \in [0, 1]$.

The merging process and the idf measure are irrelevant for phrases belonging to the components of $\mathcal{F}$, since the vertex set of each component of $\mathcal{F}$ is mutually disjoint with the vertex set of any other component in $\mathcal{C}$.

6.3 Co-clustering sub-phrase-tables

The aim of this section is to generate separate clusters for the source and target phrases of each sub-phrase-table (component) $C \in \mathcal{C}$. For this purpose the Information-Theoretic Co-Clustering (ITC) algorithm [44] is employed, which is a general principled clustering algorithm that generates *hard* clusters (i.e., every element belongs to exactly one cluster) of two interdependent quantities and is known to perform well on high-dimensional and sparse data [1, 135]. In our case, the interdependent quantities are the source and target phrases and the sparse data is the sub-phrase-table.

ITC is a search algorithm similar to K-means, in the sense that a cost function is minimized at each iteration step and the number of clusters for both quantities are meta-parameters. The number of clusters is set to the most conservative initialization for both source and target phrases, namely to as many clusters as there are phrases. At each iteration, new clusters are constructed based on the identification of the argmin of the cost function for each phrase, which gradually reduces the number of clusters.

We observe that conservative choices for the meta-parameters often result in good paraphrases being in different clusters. To overcome this problem, the hard clusters are converted into soft (i.e., an element may belong to several clusters): One step before the stopping criterion is met, we modify the algorithm so that instead of assigning a phrase to the cluster with the smallest cost we select the bottom-X clusters ranked by cost. Additionally, only a certain number of phrases is chosen for soft clustering.

Both selections are done conservatively with criteria based on the properties of the cost functions.

The formation of clusters leads to a natural refinement of the idf measure defined in eqn. (6.2): The quantity $c(x, x')$ is redefined as the number of components in which the phrases x and x' co-occur in at least one cluster.

6.4 Monolingual graphs

We proceed with converting the clusters into directed, weighted graphs and then extract paraphrases for both the source and target side. For brevity we explain the process restricted to the source clusters of a sub-phrase-table, but the same method applies for the target side and for all sub-phrase-tables in the collection $\mathcal{C}$.

Each source cluster is converted into a graph G as follows: The vertex set consists of the phrases of the cluster and an edge between s and s' exists, if (a) s and s' have at least one translation from the same target cluster, and (b) $idf(s, s')$ is greater than some threshold σ. If two phrases that satisfy condition (b) and have translations in more than one common target cluster, a distinct such edge is established. All edges are bi-directional with distinct weights for both directions.

Figure 6.3 depicts an example of such a construction; a link between a phrase s_i and a target cluster implies the existence of at least one translation for s_i in that cluster. We are not interested in the target phrases and they are thus not shown. For simplicity we assume that condition (b) is always satisfied and the extracted graph contains the maximum possible edges. Observe that phrases s_3 and s_4 have two edges connecting them, (due to target clusters T_c and T_d) and that the target cluster T_a is irrelevant to the construction of the graph, since s_1 is the only phrase with translations in it. This

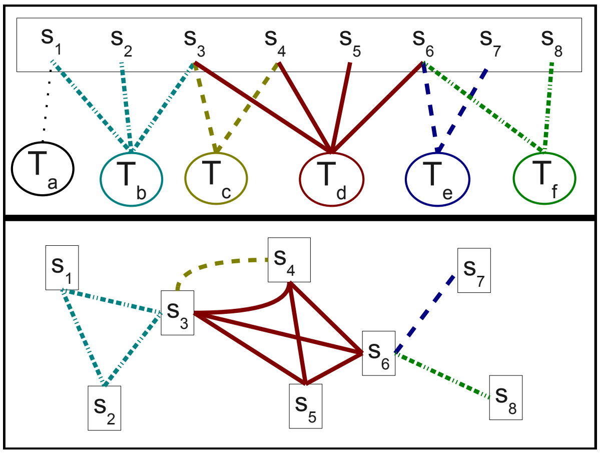


Figure 6.3: Top: A source cluster containing phrases s_1,..., s_8 and the associated target clusters T_a,..., T_f. Bottom: The extracted graph from the source cluster. All edges are bi-directional.

conversion of a source cluster into a graph G results in the formation of subgraphs in G, where each subgraph is generated by a target cluster. In general, if condition (b) is not always satisfied, then G need not be connected and each connected component is treated as a distinct graph.

Analogous to KB, we introduce *feature* vertices to G: For each phrase vertex s, its part-of-speech (POS) tag sequence and stem sequence are identified and inserted into G as new vertices with bi-directional weighted edges connected to s. If phrase vertices s and s' have the same POS tag sequence, then they are connected to the same POS tag feature vertex. Similarly for stem feature vertices. See Figure 6.4 for an example. Note that we do not allow edges between POS tag and stem feature vertices. The purpose of the feature vertices, unlike KB, is primarily for smoothing and secondarily for identifying paraphrases with the same syntactic information and this will become clear in the description of the computation of weights.

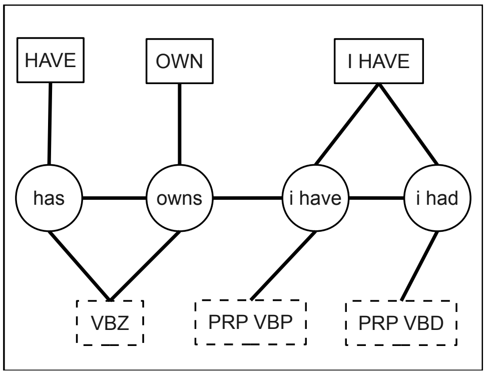


Figure 6.4: Adding feature vertices to the extracted graph $(\texttt{has}) \rightleftharpoons (\texttt{owns}) \rightleftharpoons (\texttt{i have}) \rightleftharpoons (\texttt{i had})$, where $\rightleftharpoons$ denotes a bi-directional edge. Phrase, POS tag feature and stem feature vertices are drawn in circles, dotted rectangles and solid rectangles respectively. All edges are bi-directional.

The set of all phrase vertices that are adjacent to s is written as $\Gamma(s)$, and referred to as the *neighborhood* of s. Let $n(s,t)$ denote the co-occurrence count of a phrase-table entry (s,t). We define the strength of s in the subgraph generated by cluster T as

$$n(s;T) = \sum_{t \in T} n(s,t), \tag{6.3}$$

which is simply a partial occurrence count for s. We proceed with computing weights for all edges of G:

Phrase$\rightleftharpoons$phrase weights: Inspired by the notion of *preferential attachment* [160], which is known to produce power-law weight distributions for evolving weighted networks [9], we set the weight of a directed edge from s to s' to be proportional to the strengths of s' in all subgraphs in which both s and s' are members. Thus, in the random walk framework, s is more likely to visit a stronger (more reliable) neighbor. If $T_{s,s'} = \{T \; | s$ and s' coexist in subgraph generated by $T\}$, then the weight $w(s \to s')$ of the directed edge from s to s' is given by

$$w(s \to s') = \sum_{T \in T_{s,s'}} n(s'; T), \tag{6.4}$$

if $s' \in \Gamma(s)$ and 0 otherwise.

Phrase⇌feature weights: As mentioned above, feature vertices have the dual role of carrying syntactic information and smoothing. From eqn. (6.4) it can be deduced that, if for a phrase s, the number of its outgoing weights is close to the number of its incoming weights, then this is an indication that at least a significant part of its neighborhood is reliable; the larger the strengths, the more certain the indication. Otherwise, either s or a significant part of its neighborhood is unreliable. The amount of weight from s to its feature vertices should depend on this observation and we thus let

$$\text{net}(s) = \left| \sum_{s' \in \Gamma(s)} (w(s \to s') - w(s' \to s)) \right| + \epsilon, \tag{6.5}$$

where ϵ prevents net(s) from becoming 0 (see Section 6.6.1 for an appropriate choice for ϵ). The net weight of a phrase vertex s is distributed over its feature vertices as

$$w(s \to f_X) \; = < w(s \to s') > \; + \; \text{net}(s), \tag{6.6}$$

where the first summand is the average weight from s to its neighboring phrase vertices and $X = \text{POS}, \text{STEM}$. If s has multiple POS tag sequences, we distribute the weight of eqn. (6.6) relative to the co-occurrences of s with the respective POS tag feature vertices. The quantity $< w(s \to s') >$ accounts for the basic smoothing and is augmented by a value net(s) that measures the reliability of s's neighborhood; the more unreliable the neighborhood, the larger the net weight and thus larger the overall weights to the feature vertices.

The choice for the opposite direction is trivial:

$$w(f_X \to s) = \frac{1}{|\{s' : (f_X, s') \text{ is an edge }\}|}, \tag{6.7}$$

where $X = \text{POS}, \text{STEM}$. Note the effect of eqns. (6.5)–(6.7) in the case where the neighborhood of s has unreliable strengths: In a random walk the feature vertices of s will be preferred and the resulting similarities between s and other phrase vertices will be small, as desired. Nonetheless, if the syntactic information is the same with any other phrase vertex in G, then the paraphrases will be captured.

The transition probability from *any* vertex u to *any* other vertex v in G, i.e., the probability of hopping from u to v in one step, is given by

$$p(u \to v) = \frac{w(u \to v)}{\sum_{v'} w(u \to v')}, \tag{6.8}$$

where we sum over all vertices adjacent to u in G. We can thus compute the similarity between *any* two vertices u and v in G by their commute time, i.e., the expected number of steps in a round trip, in a random walk from u to v and then back to u, which is denoted by $\kappa(u, v)$. Since $\kappa(u, v)$ is a distance measure, the smaller its value, the more similar u and v are. In Section 6.6.1 a method for the computation of κ is described.

6.5 Converting distances into counts

The distance $\kappa(u, v)$ of a vertex pair u, v in a graph G is converted into a co-occurrence count $n_G(u, v)$ with a novel technique: In order to assess the quality of pair u, v with respect to G we compare $\kappa(u, v)$ with $\kappa(u, x)$ and $\kappa(v, x)$ for all other vertices x in G. We thus consider the average distance of u with the other vertices of G other than v, and similarly for v. This quantity is denoted by $d(u, G - v)$ and $d(v, G - u)$ respectively, and by definition it is given by

$$d(i, G - j) \;=\; \sum_{\substack{x \in G \\ x \neq j}} \kappa(i, x)\, p_G(x|i), \qquad \text{for } i, j \in \{u, v\} \text{ and } i \neq j, \tag{6.9}$$

where $p_G(x|i) \equiv p(x|G, i)$ is a yet unknown probability distribution with respect to G. The quantity $(d(u, G - v) + d(v, G - u))/2$ can then be viewed as the average distance of the pair u, v to the rest of the graph G. The co-occurrence count of u and v in G is thus defined by

$$n_G(u, v) \;=\; (\; d(u, G - v) + d(v, G - u) \;)\, / 2\kappa(u, v). \tag{6.10}$$

In order to calculate the probabilities $p_G(\cdot|\cdot)$ we employ the following heuristic: Starting with a uniform distribution $p_G^{(0)}(\cdot|\cdot)$ at timestep $t = 0$, we iterate

$$d^{(t)}(i, G - j) \;=\; \sum_{\substack{x \in G \\ x \neq j}} \kappa(i, x)\, p_G^{(t)}(x|i), \qquad \text{for } i, j \in \{u, v\} \text{ and } i \neq j \tag{6.11}$$

$$n_G^{(t)}(u, v) \;=\; \frac{d^{(t)}(u, G - v) + d^{(t)}(v, G - u)}{2\kappa(u, v)} \tag{6.12}$$

$$p_G^{(t+1)}(v|u) \;=\; \frac{n_G^{(t)}(u, v)}{\sum_{x \in G} n_G^{(t)}(u, v)} \tag{6.13}$$

for all pairs of vertices u, v in G until convergence. Experimentally, we find that convergence is always achieved. After the execution of this iterative process we divide each count by the smallest count in order to achieve a lower bound of 1.

A pair u, v may appear in multiple graphs in the same sub-phrase-table C. The total co-occurrence count of u and v in C and the associated conditional probabilities are thus given by

$$n_C(u,v) = \sum_{G \in C} n_G(u,v) \tag{6.14}$$

$$p_C(v|u) = \frac{n_C(u,v)}{\sum_{x \in C} n_C(u,x)}. \tag{6.15}$$

A pair u, v may appear in multiple sub-phrase-tables and for the calculation of the final count $n(u,v)$ we need to average over the associated counts from all sub-phrase-tables. Moreover, we have to take into account the type of the vertices: For the simplest case where both u and v represent phrase vertices, their expected count is, by definition, given by

$$n(s,s') = \sum_C n_C(s,s') p(C|s,s'). \tag{6.16}$$

On the other hand, if at least one of u or v is a feature vertex, then we have to consider the phrase vertex that generates this feature: Suppose that u is the phrase vertex s=‘acquire’ and v the POS tag vertex f=‘NN’ and they co-occur in two sub-phrase-tables C and C' with positive counts $n_C(s,f)$ and $n_{C'}(s,f)$ respectively; the feature vertex f is generated by the phrase vertices ‘ownership’ in C and by ‘possession’ in C'. In that case, an interpolation of the counts $n_C(s,f)$ and $n_{C'}(s,f)$ as in eqn. (6.16) would be incorrect and a direct sum $n_C(s,f) + n_{C'}(s,f)$ would provide the true count. As a result we have

$$n(s,f) = \sum_{s'} \sum_{C} n_C(s, f(s'))\, p(C|s, f(s')), \tag{6.17}$$

where the first summation is over all phrase vertices s' such that $f(s') = f$. With a similar argument we can write

$$n(f,f') = \sum_{s,s'} \sum_{C} n_C(f(s), f(s'))\, p(C|f(s), f(s')). \tag{6.18}$$

For the interpolants, from standard probability we find

$$p(C|u,v) = \frac{p_C(v|u)p(C|u)}{\sum_{C'} p_{C'}(v|u)p(C'|u)}, \tag{6.19}$$

where the probabilities $p(C|u)$ can be computed by considering the likelihood function

$$\ell(u) = \prod_{i=1}^{N} p(x_i|u) = \prod_{i=1}^{N} \sum_{C} p_C(x_i|u)p(C|u)$$

and by maximizing the average log-likelihood $\frac{1}{N}\log \ell(u)$, where N is the total number of vertices with which u co-occurs with positive counts in all sub-phrase-tables.

Finally, the desired probability distributions are given by the relative frequencies

$$p(v|u) = \frac{n(u,v)}{\sum_x n(u,x)}, \tag{6.20}$$

for all pairs of vertices u, v.

6.6 Experiments

6.6.1 Setup

The data for building the phrase-table P is drawn from DE-EN bitexts crawled from `www.project-syndicate.org`, which is a standard resource provider for the WMT campaigns (News Commentary bitexts, see, e.g. [22]). The bitext consists of 125K sentences; word alignment was performed running GIZA++ in both directions and generating the symmetric alignments using the 'grow-diag-final-and' heuristics. The resulting P has 7.7M entries, 30% of which are '1-1', i.e., entries (s,t) that satisfy $p(s|t) = p(t|s) = 1$. These entries are irrelevant for paraphrase harvesting for both the baseline and our method, and are thus excluded from the process.

The initial giant component P_0 contains 1.7M vertices (Figure 6.5), of which 30% become residues and are used to construct R. At each iteration of the decomposition of a giant component, we remove the top $0.5\% \cdot size$ cut-vertices ranked by degree of connectivity, where $size$ is the number of vertices of the giant component and set $\theta = 2500$ as the stopping criterion. The latter choice is appropriate for the subsequent step of co-clustering the components, for both time complexity and performance of the ITC algorithm. In the components emerging from the decomposition of R_0, we observe an excessive number of cut-vertices. Note that vertices that consist in these components can be of two types: i) former residues, i.e., residues that emerged from the decomposition of P_0, and ii) other vertices of P_0. Cut-vertices can be of either type. For each component, we remove cut-vertices that are not translations of the former residues of that component. Following this pruning strategy, the degeneracy of excessive cut-vertices does not reappear in the subsequent iterations of decomposing components generated by new residues, but the emergence of two giant components was observed: One consisting mostly of source type vertices and one of target type vertices. Without going into further details, the algorithm can extend to multiple giant components straightforwardly. For the merging process of the collection $\mathcal{D}$ we set $\delta = 5000$, to avoid the emergence of a giant component. The sizes of the resulting sub-phrase-tables are shown in Figure 6.6. For the ITC algorithm we use the smoothing technique discussed in [45] with $\alpha = 10^6$.

For the monolingual graphs, we set $\sigma = 0.65$ and discard graphs with more than 20 phrase vertices, as they contain mostly noise. Thus, the sizes of the graphs allow us

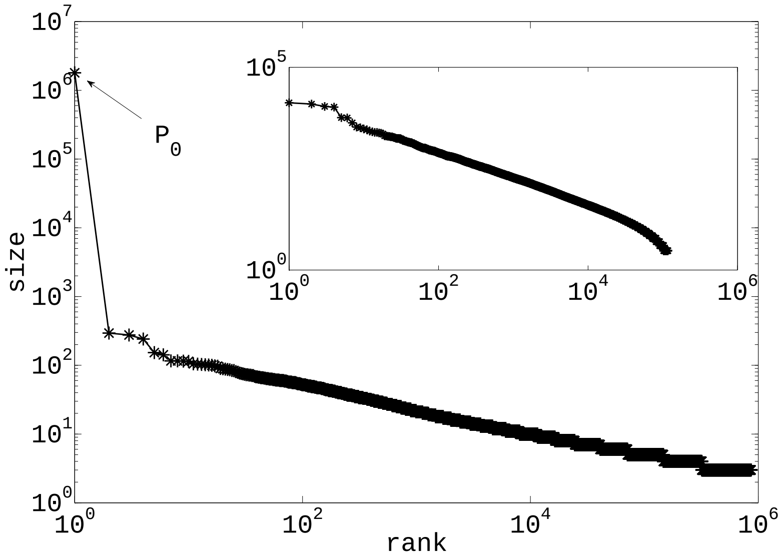


Figure 6.5: Log-log plot of ranked components according to their size (number of source and target phrases) for: Components extracted from P, '1-1' components are not shown (main). Components extracted from the decomposition of P_0 (inset).

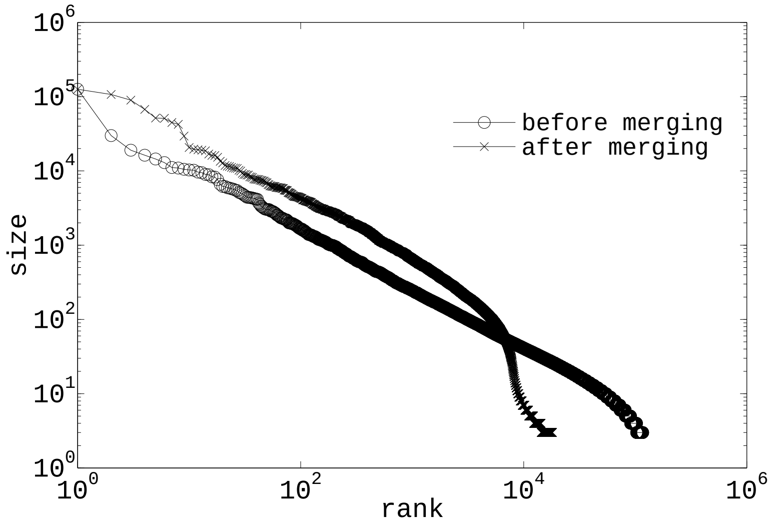


Figure 6.6: Log-log plot of ranked sub-phrase-tables according to their size (number of source and target phrases).

to use analytical methods to compute the commute times: For a graph G, we form the *transition matrix* Q, whose entries $Q(u, v)$ are given by eqn. (6.8), and the *fundamental matrix* [17, 61]

$$Z = (I - Q + \mathbf{1}\pi^T)^{-1}, \tag{6.21}$$

where I is the identity matrix, $\mathbf{1}$ denotes the vector of all ones and π is the vector of *stationary probabilities* [2], which is such that $\pi^T Q = \pi^T$ and $\pi^T \mathbf{1} = 1$, and can be computed as in [71]. The commute time between any vertices u and v in G is then given by [61]

$$\kappa(u,v) = \frac{Z(v,v) - Z(u,v)}{\pi(v)} + \frac{Z(u,u) - Z(v,u)}{\pi(u)}. \tag{6.22}$$

For the parameter of eqn. (6.5), an appropriate choice is $\epsilon = |\Gamma(s)| + 1$; for reliable neighborhoods, this quantity is insignificant. POS tags and lemmata are generated with TreeTagger.[1]

Figure 6.7 depicts the most basic type of graph that can be extracted from a cluster; it includes two source phrase vertices a, b, of different syntactic information. Suppose that both a and b are highly reliable with strengths $n(a;T) = n(b;T) = 40$, for some target cluster T. The resulting conditional probabilities adequately represent the proximity of the involved vertices. On the other hand, the range of the co-occurrence counts is not compatible with that of the strengths. This is because i) there are no phrase vertices with small strengths in the graph, and ii) eqn. (6.10) is essentially a comparison between a pair of vertices and the rest of the graph. To overcome this problem *inflation* vertices i_a and i_b of strength 1 with accompanying feature vertices are introduced to the graph. Figure 6.8 depicts the new graph, where the lengths of the edges represent the magnitude of commute times. Observe that the quality of the probabilities is preserved but the counts are inflated, as required.

In general, if a source phrase vertex s has at least one translation t such that $n(s,t) \geq 3$, then a triplet $(i_s, f(i_s), g(i_s))$ is added to the graph as in Figure 6.8. The inflation vertex i_s establishes edges with all other phrase and inflation vertices in the graph and weights are computed as in Section 6.4. The pipeline remains the same up to eqn. (6.14), where all counts that include inflation vertices are ignored.

6.6.2 Results

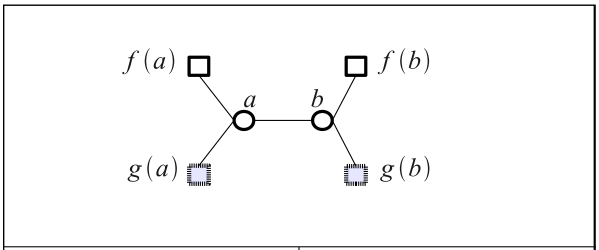
Our method generates conditional probabilities for any pair chosen from {phrase, POS sequence, stem sequence}, but for this evaluation we restrict ourselves to phrase pairs. For a phrase s, the quality of a paraphrase s' is assessed by

$$P(s'|s) \propto p(s'|s) + p(f_1(s')|s) + p(f_2(s')|s), \tag{6.23}$$

where $f_1(s')$ and $f_2(s')$ denote the POS tag sequence and stem sequence of s' respectively. All three summands of eqn. (6.23) are computed from eqn. (6.20). The baseline is given by pivoting [6],

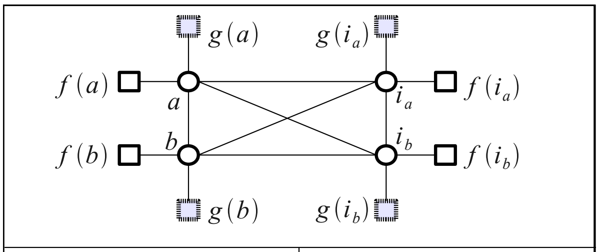
$$P(s'|s) = \sum_t p(t|s)p(s'|t), \tag{6.24}$$

[1] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/



$n(a,b)$	$=2.0$	$p(b\|a)$	$=.20$
$n(a,f(a))$	$=2.6$	$p(f(a)\|a)$	$=.27$
$n(a,g(a))$	$=2.6$	$p(g(a)\|a)$	$=.27$
$n(a,f(b))$	$=1.3$	$p(f(b)\|a)$	$=.13$
$n(a,g(b))$	$=1.3$	$p(g(b)\|a)$	$=.13$

Figure 6.7: Top: A graph with source phrase vertices a and b, both of strength 40, with accompanying distinct POS sequence vertices $f(\cdot)$ and stem sequence vertices $g(\cdot)$. Bottom: The resulting co-occurrence counts and conditional probabilities for a.



$n(a,b)$	$=11.3$	$p(b\|a)$	$=.22$
$n(a,f(a))$	$=13.5$	$p(f(a)\|a)$	$=.26$
$n(a,g(a))$	$=13.5$	$p(g(a)\|a)$	$=.26$
$n(a,f(b))$	$=6.7$	$p(f(b)\|a)$	$=.13$
$n(a,g(b))$	$=6.7$	$p(g(b)\|a)$	$=.13$

Figure 6.8: The inflated version of Figure 6.7.

where $p(t|s)$ and $p(s'|t)$ are the phrase-based relative frequencies of the translation model [86].

We select 150 phrases (an equal number for unigrams, bigrams and trigrams), for which we expect to see paraphrases, and keep the top-10 paraphrases for each phrase, ranked by the above measures. We follow [87, 102] in the evaluation of the extracted paraphrases: Each phrase-paraphrase pair is manually annotated with the following options: 0) Different meaning; 1) (i) Same meaning, but potential replacement of the phrase with the paraphrase in a sentence ruins the grammatical structure of the sentence. (ii) Tokens of the paraphrase are morphological inflections of the phrase's tokens. 2) Same meaning. Although useful for SMT purposes, 'super/substrings of' are annotated with 0 to achieve an objective evaluation.

Both methods are evaluated in terms of the Mean Expected Precision (MEP) at k; the Expected Precision for each selected phrase s at rank k is computed by $E_s[p@k] = \frac{1}{k}\sum_{i=1}^{k} p_i$, where p_i is the proportion of positive annotations for item i. The desired metric is thus given by MEP@$k = \frac{1}{150}\sum_s E_s[p@k]$. The contribution to p_i can be restricted to perfect paraphrases only, which leads to a strict strategy for harvesting paraphrases. Table 6.1 summarizes the results of our evaluation and we deduce that our method can lead to improvements over the baseline.

Method	Lenient MEP			Strict MEP		
	@1	@5	@10	@1	@5	@10
Baseline	.58	.47	.41	.43	.33	.28
Graphs	.72	.61	.52	.53	.40	.33

Table 6.1: Mean Expected Precision (MEP) at k under lenient and strict evaluation criteria.

An important accomplishment of our method is that the distribution of counts $n(u, v)$, (as given by eqns. (6.16)–(6.18)) for all vertices u and v, belongs to the power-law family (Figure 6.9). This is evidence that the monolingual graphs can simulate the phrase extraction process of a monolingual parallel corpus. Intuitively, we may think of the German side of the DE–EN parallel corpus as the 'English' approximation to an 'EN'–EN parallel corpus, and the monolingual graphs as the word alignment process.

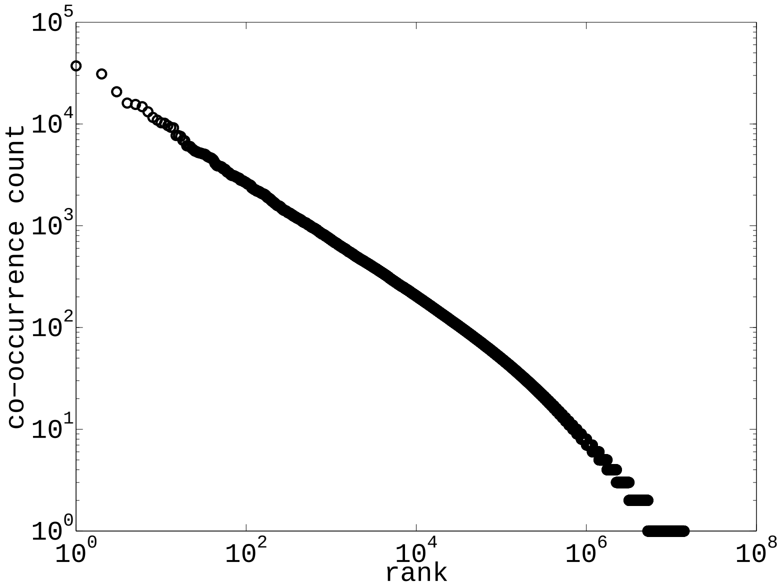


Figure 6.9: Log-log plot of ranked pairs of English vertices according to their counts

6.7 Conclusions

In this chapter we have extended graph-based methods that extract paraphrases from the phrase-table. For a source phrase that exists in the phrase-table, its paraphrases together with associated probabilities have been identified by exploiting the connectivity properties of the phrase-table. More precisely, the following research question was addressed in this chapter:

RQ7 *How should one extend the work of Kok and Brockett [87] in order to identify less noisy pairs of paraphrases and to develop a method that constructs artificial co-occurrence counts for these pairs?*

The phrase-table was decomposed into a collection of sub-phrase-tables by identifying its connected components, which represent a natural phrase pair clustering of the phrase-table. This is because each connected component has its own attributes regarding vocabulary, syntax and word-ordering of the parallel corpus. The largest connected component, however, was found to be of significant size, i.e., its number of entries were comparable to those of the initial phrase-table. This happens because very frequent source or target phrases have a relatively huge number of translations and are thus responsible for the formation of a giant cluster. These phrases operate as hubs that connect unrelated parts of the giant component. Their removal thus reveals those clusters that then form new sub-phrase-tables. This operation of hub removal was performed iteratively until the largest emerging connected component stopped being of significant size. All removed phrases were appropriately inserted into sub-phrase-tables, and a conservative merging of sub-phrase-tables was performed in order to a) mitigate the risk of good paraphrased pairs being in diffrent sub-phrase-tables, and b) reduce the size of the total collection. An important outcome of this construction is that a phrase may appear in multiple sub-phrase-tables.

By exploiting connectivity properties of each sub-phrase-table, separate source phrase and target phrase clusters were generated. This was achieved with a co-clustering algorithm that is based on Information Theory. Each cluster, of either source or target phrases, gave rise to a monolingual weighted graph, whose structure was inherited from its corresponding sub-phrase-table graph representation. Thus, the collection of sub-phrase-tables was transformed into two new collections: One consisting of source phrase graphs and another consisting of target phrase graphs. The rest of the process applies to both collections independently.

For each monolingual weighted graph the similarity between each pair of phrases was computed based on their commute time, which is a random walk-based distance measure. Since a pair of phrases may appear in multiple monolingual graphs we computed the expected value of their commute time. The involved probability distribution was obtained through a novel heuristic iterative method. This method also produced the conversion of commute time distances between a pair of phrases into co-occurrence counts of such a pair. These counts were interpreted as artificial co-occurrence counts between a pair of source (target) phrases in hypothetical source (target) language-to-paraphrased source (target) language aligned corpora.

Chapter 7

Computing Compositionally Aware Translation Probabilities

This chapter is devoted to answering **RQ8**. Given a phrase pair we show how to compute translation probabilities based on structure provided by its known alignments. This structure, namely bilingual chunks which partition the phrase pair, allows recursive decompositions of the phrase pair into multiple sub-phrase pairs. Our methods use basic probability theory and the assumption of a statistical form of compositionality for aligned bitexts. This assumption generalizes monolingual statistical composition and reveals the need for a formal distinction between a pair of strings and bilingual aligned chunks. Apart from the theoretical interest of our derivations, experiments with the resulting compositionally aware translation probabilities yield applications in estimating translation probabilities for unseen phrase pairs.

7.1 Introduction

Given a bitext, i.e., a collection of source language sentences and their target language translations, the first key step of SMT is the extraction of translation rules. This typically involves the identification of alignments between segments on both sides of a sentence pair, for all pairs in the bitext. Such segments can be either at the word level [19] or phrase level [95] and extensive research has been devoted in finding high quality alignments. [1]

The resulting aligned segments give rise to the smallest and most likely translation rules, which form building blocks, or components, of larger translation rules. Typically, larger translation rules are extracted from unions of components, i.e., from merging building blocks, and are not necessarily linguistically motivated.

[1] For word-based models see indicatively [42, 149] for word alignment with linguistic annotation, [20, 38] for syntax-based models, and see [4, 49, 72, 93] for discriminative word alignment. For phrase-based models, see Section 3.2.

Every translation rule is equipped with a probability which is typically calculated from the empirical distribution f of all extracted translation rules. In general, if a (source, target) phrase pair (s, t) has been identified as a translation rule, then

$$f(t|s) = \frac{n(s,t)}{\sum_{t'} n(s,t')}, \tag{7.1}$$

where $n(s, t)$ is the number of times that phrase pair (s, t) has been extracted as a translation rule from all aligned sentence pairs in the bitext. The larger the bitext, the more indicative $f(t|s)$ is of the translation quality, especially for smaller phrase pairs.

However, (7.1) overlooks the compositional nature of (s, t) and it is consequently less accurate for larger, sparser and erroneous translation rules. If word-level alignments are known, the so-called lexical weight can to some extent counterbalance this problem. This weight is essentially the product of all word-level translation probabilities from the aligned words in (s, t) [82]. Its inclusion as an additional feature during decoding, i.e., the process of translating a source string to a target string, is known to improve performance. The research question that is addressed here is the following:

RQ8 *Given an unseen phrase pair (s, t), how can one compute the translation probability $p(t|s)$ based on their most likely composition of building blocks?*

In this chapter we are interested in providing some further understanding into how local structure shapes translation probabilities, and we focus on the component level. The purpose is twofold: 1) To provide a theoretically sound framework for statistical bilingual compositionality, with components being the constituents of composition. 2) To derive translation probabilities $p(t|s)$ from that framework for applications, and in particular for estimating translation probabilities of unseen phrase pairs.

Our set up is independent of whether

- alignments are at the word or phrase level,
- phrase pair formation is linguistically motivated or not,
- phrase pairs are continuous or discontinuous.

Using basic probability theory, we show that the desired probability $p(t|s)$ is basically a sum over all possible ways of combining components of (s, t) that lead to the formation of (s, t). Each such combination carries a probability that depends on the order with which components are chosen.

7.2 Background: Phrase pair extraction

As mentioned in Section 7.1, the desired probability that combines global and local structure of a phrase pair should be independent of how alignments are computed. We are only interested in the fact that a phrase pair can be recursively decomposed into sub-phrase pairs, with each phrase pair being a legitimate translation rule. The base case of a possible recursive decomposition is, naturally, a building block of the phrase pair, i.e., a component. How this component is initially formed is indifferent to our computations.

Since word-level alignments are prevalent in SMT, we build up the definition of a translation rule from a word-aligned sentence pair. The rest of this section explains the most widely used strategy for extracting phrase pairs and is attributed to [82, 109]. We provide a slightly more general definition, and the description follows Chapter 3.

Given a word-aligned sentence pair, a *component* is a set of source and target words such that it is either an unaligned word or 1) It is possible to form a path between any two words of the component via word alignments, and 2) It is impossible to form such a path between any word in the component and any word outside the component. A component is called discontinuous if at least either its source or target words form a discontinuous substring in the source or target sentence, respectively. A component is called continuous if it is not discontinuous. Figure 7.1(a) shows an example of a word-aligned sentence pair with source and target words labeled by $a, ..., g$ and $\alpha, ..., \zeta$, respectively. Word alignments admit five components which are distinctively shown as a set in Figure 7.1(b). The component with words a, d and α is discontinuous, b and δ are unaligned and the remaining two components are continuous.

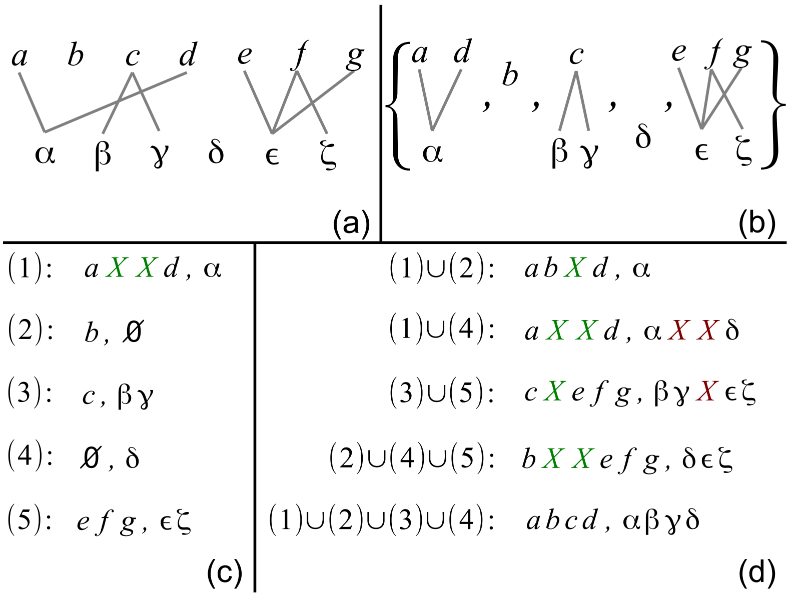


Figure 7.1: (a) A word aligned sentence pair. (b) The set of its components. (c) All phrase pairs that correspond to components. (d) Some phrase pairs that correspond to unions of components.

A phrase pair is *extracted* from a word-aligned sentence pair as a translation rule if and only if its source and target words form a component or a union of components. Thus, for a word-aligned sentence pair with n components there are $2^n - 1$ possible translation rules that can be extracted. Clearly, if the union of components is discontinuous then the extracted phrase pair is also discontinuous. In the previous example, Figure 7.1(c) shows all extracted phrase pairs that correspond to components and Figure 7.1(d) shows 5 of the possible 26 phrase pairs that correspond to unions of components. Token X is used to denote a single discontinuity.

There are several ways of filtering phrase pairs that are allowed to form translation rules. A standard criterion is complete absence of alignments, so that $(b, \emptyset)$, $(\emptyset, \delta)$ as well as (b, δ) would be disallowed. Discontinuous phrase pairs are used in hierarchical SMT [27], although only certain types of discontinuities are allowed, as determined by bilingual parsing. Without using any linguistic information, the appropriate selection of discontinuous phrase pairs and engineering aspects of their inclusion during decoding are matters of ongoing research [54, 56, 70].

Regardless of the restrictions imposed on allowed translation rules, counts of extracted phrase pairs from all sentence pairs in the bitext give rise to the empirical distribution (7.1). For discontinuous phrase pairs, token X is treated as a wild card, so that, for example,

$$n(aXXd, \alpha) = \sum_{u_1, u_2} n(au_1u_2d, \alpha),$$

where the sum is over all tokens u_1 and u_2 such that (au_1u_2d, α) is an extracted phrase pair. Consecutive gaps as in this example can be merged into a single discontinuity; we do not follow this approach in this work and opt for the general case.

For our purposes all extracted phrase pairs are allowed, with the exception of the (b, δ)-type translation rule in the example of Figure 7.1. In the next section we show how to compute translation probabilities of any phrase pair from set of all components that has been constructed during the training stage.

7.3 Compositionally aware translation probabilities

Alignments in a parallel corpus admit formation of components and we show how to exploit this structure in order to compute translation probabilities for *any* phrase pair. Our key assertion is that phrase pairs obey a statistical form of compositionality, with the constituents of composition being the components. Informally, in a monolingual setting this is akin to saying that empirical counts of monolingual building blocks (words) provide the basis for inferring knowledge about phrases. In computational linguistics such sought knowledge is typically semantics. For our purposes semantics nor any other linguistic information play any explicit role in our derivations.

As mentioned in Section 7.2, components and unions of components give rise to phrase pairs. This method of extracting translation rules from a sentence pair masks a seemingly innocent process. Strictly speaking, a component is a graph ignoring word order by definition, but carrying information about word connections. On the other hand, a phrase pair just respects the order of the sentence pair. In other words, the method of phrase pair extraction first identifies appropriate graphs from a sentence pair upon which an ordering map is applied and produces a bistring. The details of the ordering map are insignificant for our purposes. The key observation is that we have a conversion of sets into a pair of strings. In a probabilistic setting these two quantities are also distinguished, i.e., they are represented by different random variables.

7.3.1 Components as partitions of phrase pairs

Let (s,t) denote any phrase pair, i.e., a pair of source and target language strings; s consists of words from of a source language corpus and similarly for t. Let (s_c,t_c) denote a component of an aligned bitext, i.e., a pair of sets; s_c is the set of source words of component c and similarly for t_c. Assuming significant vocabulary overlap between the monolingual corpora and their bitext counterparts, then any phrase pair (s,t) can be constructed from the set of all components, say C, of the aligned bitext. This is done by identifying an appropriate subset g of C and then by applying an ordering map to g to produce (s,t).

In particular, if $g = \{(s_1,t_1),...,(s_k,t_k)\}$ is a set of k components (with arbitrary labeling), then we have

$$O(g) = O\big((s_1,t_1),...,(s_k,t_k)\big) = (s,t), \tag{7.2}$$

for some map O that collects g's source and target words and orders them to produce phrase pair (s,t). We call g the *structure* of (s,t) and write $g(s,t)$ whenever emphasis is needed. The structure g is just one possible partition of the words of (s,t), or equivalently, different structures can produce the same phrase pair (s,t). Denote the set of all such partitions that eventually yield (s,t) by

$$G(s,t) = \{g : \exists O \text{ such that } O(g) = (s,t)\}. \tag{7.3}$$

Then, for the translation probability $p(t|s)$ we naturally have

$$p(t|s) = \sum_{g\in G(s,t)} p(t,g|s). \tag{7.4}$$

We proceed with simplifying this sum.

In general, the number of ways of partitioning any set of cardinality N is given by B_N, the Nth Bell number (see Section 4.5.1). Here, we disallow complete absence of alignments in a structure so that $|G(s,t)| = B_{|s|+|t|} - B_{|s|}B_{|t|}$, where $|s|$ denotes the number of words in phrase s and similarly for $|t|$. This quantity still grows rapidly in $|s| + |t|$ but is just an upper bound for all possible structures that produce (s,t).

In practice most candidate structures would be disallowed because components of such structures would not exist in C. Empirically, a phrase pair that is extracted from an aligned bitext typically has unique structure or multiple structures with one of dominant frequency. We thus assume that for the remaining allowed structures in G only a single structure provides significant contribution to (7.4).

Although this approximation is clearly less accurate for larger $|s|$ and $|t|$, it simplifies our following computations immensely. As we shall also explain later the inclusion of more than one structure is to some extent possible. Hence, for the rest of this section we assume that

$$p(t|s) = p(t, g|s), \tag{7.5}$$

for some 'dominant' structure g of (s, t).

7.3.2 Core computations

Given a set of components C, for any $g \subseteq C$ we consider the probability

$$\begin{aligned} p(g) &= p(\{(s_1, t_1), ..., (s_k, t_k)\}) \\ &= \sum_{c=1}^{k} p(s_c, t_c), \end{aligned} \tag{7.6}$$

as a means of realizing the compositional nature of all phrase pairs that can be produced by g. Assuming that C's members satisfy $p(C) = 1$, then (7.6) is a legitimate probability. In a monolingual setting this is akin to saying that the probability of a subset of a vocabulary is given by its word probabilities; this is of course a very shallow approach and more complicated techniques are employed for understanding monolingual compositionality. In our case, however, components provide at least proto-segments for both language sides, which makes (7.6) meaningful. Here, a segment is a continuous or discontinuous chunk that is of joint maximal expressive power and minimal length within a given sentence. By proto-segment we refer to a chunk that does not meet this criterion optimally.

For any phrase pair (s, t) that can be produced by g, the combination of (7.5) and (7.6) yields

$$p(t|s) \approx \sum_{c=1}^{k} p(t, s_c, t_c|s), \tag{7.7}$$

which allows us to explore the relationship between sets of words (s_c and t_c) and strings (s and t). Using the chain rule we have

$$p(t|s) = \sum_{c=1}^{k} p(t|s_c, t_c, s)p(s_c, t_c|s). \tag{7.8}$$

The key probability that needs inspection is $p(t|s_c, t_c, s)$ which will be approximated in two steps. First, it is easier to overlook the set status of s_c and t_c and consider instead their ordered versions $\mathtt{s}$ and $\mathtt{t}$ with order determined by s and t respectively. In this case we have $p(t|\mathtt{s}, \mathtt{t}, s) = p(t \setminus \mathtt{t}\ |\mathtt{s}, \mathtt{t}, s)$, where $t \setminus \mathtt{t}$ is the remaining string from t

when its substring $\mathtt{t}$ is removed. This is true because we are already provided with the information that $\mathtt{t}$ is included in the string to be estimated. Also, since s contains $\mathtt{s}$ we have $p(t \setminus \mathtt{t}\ |\mathtt{s}, \mathtt{t}, s) = p(t \setminus \mathtt{t}\ |\mathtt{s}, \mathtt{t}, s \setminus \mathtt{s})$. Back to our unordered framework we thus consider the approximation

$$p(t|s_c, t_c, s) \approx p(t \setminus t_c|s_c, t_c, s \setminus s_c), \tag{7.9}$$

where $t \setminus t_c$ is the remaining string from t when t_c's words are removed; similarly for $s \setminus s_c$. Clearly, (7.9) is an approximation rather than equality entirely due to the lack of order in s_c and t_c.

The second step of our approximation is based on qualitative aspects of alignments in SMT and simplifies (7.9) even further. The purpose of alignments, and consequently component formation in the first place is to identify maximal associations within the smallest possible bilingual chunks. As a result, whatever exists in a component of a sentence pair has supposedly been found not to have an impact on the remaining sentence pair and vice-versa. Thus, given $s \setminus s_c$, the additional knowledge of s_c and t_c is of little value when estimating $t \setminus t_c$. In other words, we can assume that

$$p(t \setminus t_c|s_c, t_c, s \setminus s_c) \approx p(t \setminus t_c|s \setminus s_c). \tag{7.10}$$

This approximation is less accurate whenever each of $s \setminus s_c$ and $t \setminus t_c$ is at most a proto-segment, or equivalently, if t and s convey richer associations than $s \setminus s_c$ and $t \setminus t_c$. It is important to note that (7.10) is false if the aligned bitext is treated as a stochastic process [108], but this is clearly not the case here.

Inserting (7.10) in (7.8) results in

$$p(t|s) = \sum_{c=1}^{k} p(t \setminus t_c|s \setminus s_c)p(s_c, t_c|s), \tag{7.11}$$

which is in recursive form since $p(t \setminus t_c|s \setminus s_c)$ is a sub-phrase pair of (s, t) with structure $g(s \setminus s_c, t \setminus t_c)$. In other words, $p(t \setminus t_c|s \setminus s_c)$ can also be written in the form of (7.11). An iterative application of this process results in a statistically elaborate relationship of (s, t) with its structure.

Let π denote a permutation of $\{0, 1, ..., k\}$ with $\pi(0) = 0$, then (7.11) can be rewritten as

$$p(t|s) = \sum_{\pi} \prod_{c=1}^{k} p(s_{\pi(c)}, t_{\pi(c)}|s_{\pi}^{c}), \tag{7.12}$$

where

$$s_{\pi}^{c} = s \setminus \bigcup_{i=0}^{c-1} s_{\pi(i)} \quad \text{with} \quad s_0 = \emptyset \tag{7.13}$$

is the remaining substring of s at each step of a recursion. The transition from (7.11) to (7.12) is only a matter of algebra and we leave the details for Appendix B.

Using the chain rule we have

$$p(t|s) = \sum_{\pi} \prod_{c=1}^{k} p(t_{\pi(c)}|s_{\pi(c)}, s_{\pi}^{c}) p(s_{\pi(c)}|s_{\pi}^{c}), \tag{7.14}$$

where the probability $p(s_{\pi(c)}|s_{\pi}^{c})$ is next to be estimated. In order to get an intuition of how this probability is interpreted, it is easier to consider a more general case. From (7.7) it is trivial to see that $\sum_{c=1}^{k} p(s_c, t_c|s, t) = 1$. This forces $p(s_c, t_c|s, t)$ to be interpreted as the "probability of selecting component c from structure $g(s, t)$". Equivalently $p(s_c|s)$ is interpreted as the "probability of selecting set s_c from the source sets of structure $g(s, t)$". By setting this probability to be uniform everywhere in (7.14), basic calculations yield

$$p(t|s) = \frac{1}{k!} \sum_{\pi} \prod_{c=1}^{k} p(t_{\pi(c)}|s_{\pi(c)}, s_{\pi}^{c}). \tag{7.15}$$

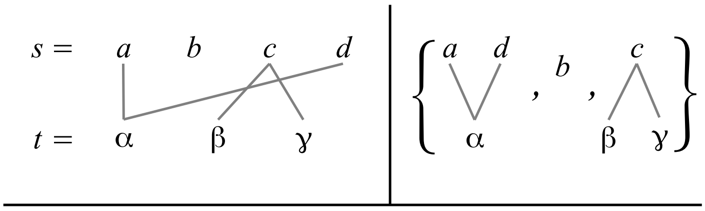
In general, for any component (σ, τ), and any source string $\mathtt{s}$ the probability $p(\tau|\sigma, \mathtt{s})$ is interpreted as follows: "Given source string $\mathtt{s}$ whose subset of words σ is known to form the source side of some component, what is the probability of observing words τ as the target side of that component in a translation of $\mathtt{s}$". This is simplified as "given source string $\mathtt{s}$, what is the probability of observing words τ as the target side of some component in a translation of $\mathtt{s}$", and write $p(\tau|\mathtt{s})$ for that approximation. This is an abuse of notation because it is possible that $p(\tau|\mathtt{s}) = p(\tau'|\mathtt{s}) = 1$ with $\tau \neq \tau'$. This happens if both τ and τ' appear always in *all* translations of $\mathtt{s}$. Nonetheless, in order to avoid the introduction of further notation we use this simplistic form and (7.15) becomes

$$p(t|s) = \frac{1}{k!} \sum_{\pi} \prod_{c=1}^{k} p(t_{\pi(c)}|s_{\pi}^{c}), \tag{7.16}$$

which completes the core of our computations.

In order to get a hands-on understanding of (7.16) we provide a fully worked example. By considering unaligned words as components and denoting the empty string (NULL) also by $\emptyset$, then for phrase pair $(s, t) = (abcd, \alpha\beta\gamma)$ with structure as in Figure 7.2 we have

$$\begin{aligned}
&p(\alpha\beta\gamma|abcd) = \\
&\quad \frac{1}{6}\big[\, p(\alpha|abcd)\; p(\emptyset|bc)\; p(\beta\gamma|c) + \\
&\qquad p(\alpha|abcd)\; p(\beta\gamma|bc)\; p(\emptyset|b) + \\
&\qquad p(\emptyset|abcd)\; p(\alpha|aXcd)\; p(\beta\gamma|c) + \\
&\qquad p(\emptyset|abcd)\; p(\beta\gamma|aXcd)\; p(\alpha|aXXd) + \\
&\qquad p(\beta\gamma|abcd)\; p(\alpha|abXd)\; p(\emptyset|b) + \\
&\qquad p(\beta\gamma|abcd)\; p(\emptyset|abXd)\; p(\alpha|aXXd)\, \big],
\end{aligned}$$



$$(s_1,t_1) = (\{a,d\},\{\alpha\})$$

$$(s_2,t_2) = (\{b\},\emptyset)$$

$$(s_3,t_3) = (\{c\},\{\beta,\gamma\})$$

$$g = \{(s_1,t_1),(s_2,t_2),(s_3,t_3)\}$$

Figure 7.2: Phrase pair $(s,t) = (abcd, \alpha\beta\gamma)$ with structure g. The labeling of components is arbitrary.

where permutations are considered in lexicographic order, so that the first line is $(1\ 2\ 3)$, the second line is $(1\ 3\ 2)$, etc. For clarity the target side of components is written as strings with order provided by $t = \alpha\beta\gamma$. Similarly the first three summands for the other direction are

$$\begin{aligned}
&p(abcd|\alpha\beta\gamma) = \\
&\frac{1}{6}\big[\,p(aXXd|\alpha\beta\gamma)\;p(b|\beta\gamma)\;p(c|\beta\gamma)+ \\
&\quad p(aXXd|\alpha\beta\gamma)\;p(c|\beta\gamma)\;p(b|\emptyset)+ \\
&\quad p(b|\alpha\beta\gamma)\;p(aXXd|\alpha\beta\gamma)\;p(c|\beta\gamma) + \ ...\,\big].
\end{aligned}$$

Simpler translation probabilities can be derived from (7.5) and (7.14) and this is shown in the next section.

7.3.3 Further approximations

From (7.5) we also have $p(t|s) = p(t|s,g)p(g|s)$, for which we can assume $p(t|s,g) \approx 1$. This is a valid approximation because g provides all information for the target string to be estimated apart from the exact ordering of its words. We thus have

$$\begin{aligned} p(t|s) \approx p(g|s) &= \sum_{c=1}^{k} p(s_c, t_c|s) \\ &= \sum_{c=1}^{k} p(t_c|s_c, s)p(s_c|s), \\ &= \frac{1}{k}\sum_{c=1}^{k} p(t_c|s_c, s), \end{aligned} \tag{7.17}$$

where at the final step $p(s_c|s)$ was set to be uniform. For the summands of (7.17) we can either consider $p(t_c|s_c, s) \approx p(t_c|s)$ or $p(t_c|s_c, s) \approx p(t_c|s_c)$ as a further approximation. It will become evident in Section 7.4 that the former is more reliable. However, it is of little value when computing $p(t|s)$ for unseen phrase pairs, when the source string in particular is unseen. We thus consider

$$p(t|s) = \frac{1}{k}\sum_{c=1}^{k} p(t_c|s_c). \tag{7.18}$$

An equally simple translation probability can be derived from (7.14). By setting $p(t_{\pi(c)}|s_{\pi(c)}, s_{\pi}^{c}) \approx p(t_{\pi(c)}|s_{\pi(c)})$ everywhere in (7.14), and regardless of what $p(s_{\pi(c)}|s_{\pi}^{c})$ is, it is easy to show that

$$p(t|s) = \prod_{c=1}^{k} p(t_c|s_c). \tag{7.19}$$

In order to compute $p(t|s)$ as in (7.16), (7.18) and (7.19) we need to estimate $p(\tau|\mathsf{s})$ and $p(\tau|\sigma)$ for any source string s and for any source (target) side σ (τ) of a component from training data.

7.3.4 Estimating probabilities from training data

Maximum likelihood estimation for $p(\tau|\mathsf{s})$ is difficult and we thus consider a basic counting scheme. Our training data consists of all extracted phrase pairs together with their structures. The example of Section 7.3.2 guides us as to how counts should be collected.

In (7.16), a phrase pair uses information from all its sub-phrase pairs, but counting each of its (source substring, target side of component) occurrence is wrong: For the phrase pair $(abcd, \alpha\beta\gamma)$ in our example, suppose we count for instance the occurrence of $(aXXd, \{\alpha\})$. But *phrase pair* $(aXXd, \alpha)$ will also be extracted as a legitimate translation rule, which would undesirably yield an additional count for $(aXXd, \{\alpha\})$.

Thus, for each extracted phrase pair $(\mathtt{s}, \mathtt{t})$ with structure $\{(s_1, t_1), ..., (s_k, t_k)\}$ we simply count the occurrence of each of $(\mathtt{s}, t_1)$,..., and $(\mathtt{s}, t_k)$ only. This guarantees coverage of all sought information with no duplications.

By collecting all such counts from the training data we compute

$$p(\tau|\mathtt{s}) = f(\tau|\mathtt{s}) = \frac{n(\mathtt{s}, \tau)}{\sum_{\mathtt{t}} n(\mathtt{s}, \mathtt{t})}, \tag{7.20}$$

for any source string $\mathtt{s}$ and any target side τ of a component. The numerator counts how many times τ has been seen in all translation of $\mathtt{s}$ and the denominator is the occurrence count of $\mathtt{s}$.

The estimation of $p(\tau|\sigma)$ for any source (target) side σ (τ) of a component seems clearer: MLEs can be computed based on the set of all components C. However, in preliminary experiments we found that a smoother distribution was needed. Thus, instead of C we consider the set of all extracted phrase pairs. I.e., the ordered version of (σ, τ) is considered and $p(\tau|\sigma)$ is computed as in (7.1).

For any phrase pair (s, t), we have so far assumed a unique, dominant structure g. In order to accommodate $p(t|s)$ for multiple structures, we consider linear interpolation

$$p(t|s) = \sum_{g \in G(s,t)} \lambda(g) p(t, g|s), \tag{7.21}$$

with $\lambda(g) = n(g) / \sum_{g' \in G(s,t)} n(g')$, where $n(g)$ is the frequency of g in the aligned bitext. A comparison of (7.4) with (7.21) deems $\lambda(g)$ redundant. Without going into details, it is a necessary remedy that balances the choice for $p(s_c|s)$ earlier, when a single structure was assumed.

Most of our efforts in collecting counts revolved around unaligned words. Equations (7.16), (7.18) and (7.19) do not suggest special treatment for unaligned words. In practice we found that treating an unaligned word as component was not always the optimal case. Given structure $g(s, t)$, let $x = \{x_1, ..., x_l\}$ and $y = \{y_1, ..., y_m\}$ denote the sets of unaligned source and target words respectively. In preliminary experiments the following two approaches were found to perform best:

NoNULL Remove all components $(\{x_i\}, \emptyset)$ and $(\emptyset, \{y_j\})$ from $g(s, t)$ to get new structure $g'(s, t)$. Use $g'(s, t)$ to compute (7.16) but replace each of $p(\tau|\mathtt{s})$ with $p(\tau \cup y|\mathtt{s} \vee x)$, where $\mathtt{s} \vee x$ is the substring of s with words from $\mathtt{s}$ and x. Similarly in (7.18) and (7.19) replace $p(\tau|\sigma)$ with $p(\tau \cup y|\sigma \cup x)$. Consequently, $\emptyset$ never appears in the probabilities.

TrgNULL Similar to NoNULL but remove only components $(\emptyset, \{y_j\})$ from the structure. Then replace $p(\tau|\mathtt{s})$ with $p(\tau \cup y|\mathtt{s})$ and $p(\tau|\sigma)$ with $p(\tau \cup y|\sigma)$ everywhere in (7.16) and (7.18), (7.19) respectively. In this case, $\emptyset$ appears only at the target side.

7.4 Experiments

For phrase pair (s,t) with structure $g(s,t) = \{(s_1,t_1),...,(s_k,t_k)\}$ we evaluate the performance of the following translation probabilities

$$\texttt{Permute}(t|s) = \frac{1}{k!}\sum_{\pi}\prod_{c=1}^{k} f(t_{\pi(c)}|s_{\pi}^{c}), \tag{7.22}$$

$$\texttt{Average}(t|s) = \frac{1}{k}\sum_{c=1}^{k} f(t_c|s_c), \text{ and} \tag{7.23}$$

$$\texttt{Product}(t|s) = \prod_{c=1}^{k} f(t_c|s_c), \tag{7.24}$$

where $f(\tau|\text{s})$ and $f(\tau|\sigma)$ are given by (7.20) and (7.1) respectively, and s_{π}^{c} as defined in (7.13). Each translation probability (for both directions) is computed for a subset of extracted phrase pairs from the training data and (7.21) is used wherever necessary. These three translation probabilities are compared to each other but also evaluated against the baseline which is given by $f(t|s)$ as in (7.1).

The decoder handles only continuous phrase pairs, i.e., phrase pairs are not allowed to have gaps (although their constituents are allowed to be discontinuous). Both the baseline and our system are thus standard phrase-based SMT systems [82].

Bidirectional word alignments are generated with GIZA++ [111] and 'grow-diag-final-and'. These are used to construct a phrase-table with translation probabilities, lexical weights and a reordering model with monotone, swap and discontinuous orientations, conditioned on both the previous and the next phrase. 4-gram interpolated language models with Kneser-Ney smoothing are built with SRILM [144]. A distortion limit of 6 and a phrase penalty are also used. All model parameters are tuned with MERT [110]. Decoding during tuning and testing is done with Moses [85]. All tables are significance filtered with parameter $\alpha + \epsilon$ [74].

The maximum length of each phrase in a phrase pair in the baseline is 7. For our purposes, the quantity of interest is $|g(s,t)|$. By default this is also 7, which yields 7!=5040 maximum number of permutations in (7.22). To allow faster computations we only consider phrase pairs (s,t) that satisfy $|g(s,t)| \leq 6$. For fair comparisons this further pruning is also done in all tables. No effect in the baseline's translation quality was observed, which is in line with the observations in [82].

Datasets are from the WMT'13 translation task [14]: Translation and reordering models are trained on Czech–English and German–English corpora (Table 1). Language models are trained on 35M Czech, 50M German and 94M English sentences from Europarl Corpus and News Commentary set. Tuning is done on newstest2010 and performance is evaluated on newstest2009, newstest2011 and newstest2012 with the metric BLEU [115].

BLEU scores are reported in Tables 7.2 and 7.3 for German–English and Czech–English, respectively. Each entry in the column named 'Feature' indicates which trans-

	Cz–En	De–En
Europarl (v7)	642,505	1,889,791
News Commentary (v8)	139,679	177,079
Total	782,184	2,066,870

Table 7.1: Number of filtered parallel sentences for Czech–English and German–English.

Feature	German→English				English→German			
	’09	’11	’12	Loss	’09	’11	’12	Loss
f	20.80	21.55	22.05	–	15.45	16.10	16.55	–
`Average`	20.20	20.90	21.35	0.65	14.75	15.65	16.00	0.53
`Product`	20.25	20.85	21.60	0.57	14.80	15.60	15.85	0.62
`Permute`	20.70	21.25	21.80	**0.22**	15.30	15.90	16.45	**0.15**

Table 7.2: System evaluation for 9.8M German–English phrase pairs. Method TrgNULL is used for unaligned words.

lation probability is used in the phrase-table. Column ‘Loss’ shows the average drop in BLEU when each of our systems for a particular language direction is compared with the baseline. The NoNULL and TrgNULL approach is used for Czech–English and German–English respectively, as these were the approaches that performed best for each language pair (for both directions). Systems `Average` and `Product` perform poorly when compared to f and always worse than `Permute`. The results for `Permute` when compared to f are encouraging and lead the way for computing translation probabilities for unseen phrase pairs.

7.5 Towards decoding with ad hoc translation probabilities

A method that handles unseen phrase pairs successfully during decoding is guaranteed not only to absorb the insignificant losses of `Permute` in Section 7.4, but to provide much higher translation quality. `Permute` can be valuable when generating new phrase pairs with accompanying translation probabilities. Given a test sentence the following sketch of a method is both realistic and promising.

First, segment the test sentence as there are qualitative and quantitative benefits. It decreases the number of candidate source strings to be inspected and provides initial information of what the source side of components should be. String s consisting of $s^1...s^k$ is a definite candidate string, if each s^i appears in the phrase-table. Top-n translations for each s^i are selected, thus producing n^k possible structures. Clearly, without

Feature	Czech→English				English→Czech			
	’09	’11	’12	Loss	’09	’11	’12	Loss
f	20.20	22.30	20.50	–	15.40	16.60	14.65	–
`Average`	20.00	21.70	20.20	0.37	15.00	16.30	14.40	0.32
`Product`	19.85	21.95	20.20	0.33	15.15	16.40	14.45	0.22
`Permute`	20.15	22.15	20.40	**0.10**	15.30	16.75	14.70	**−0.03**

Table 7.3: System evaluation for 3.6M Czech–English phrase pairs. Method NoNULL is used for unaligned words.

segmenting the test sentence, the precision/recall balance will be hard to achieve.

Quick pruning can then be done using `Average` or `Product`. For the remaining structures, it is the task of a target side language model and `Permute` to decide for the best performing target string t (i.e., a mini decoding step). For the resulting phrase pair, translation probabilities are obviously given by `Permute`, and lexical weights as well as reordering orientations can be trivially recovered.

In contrast with `Average` and `Product`, `Permute` uses multi granular information from a given phrase pair. If certain (source substring, target side of component) pairs are unseen , then either simple heuristics or existing smoothing techniques should be used. This is not necessarily a disadvantage for `Permute`: The more unseen such information for a candidate phrase pair (s, t) is observed, the greater the indication that (s, t) is erroneous or unnecessary.

Finally, all probabilities of this work, naturally perform best if: either bitext alignments are of phrase-level, or components of a word-aligned bitext have been post-consolidated with some bilingual chunking method. The reason is that our key assertion of statistical compositionality, namely (7.6), has to be robust. Robustness in (7.6) warrants (7.10) to be a reasonable approximation. A bad approximation for (7.10) is clearly more likely to appear if C is constructed from raw word-aligned components.

7.6 Related work

Exploitation of component structure is the key aspect of (bilingual) N–gram-based SMT [97, 121]. In this setting, the sequence of components (tuples/minimal translation units) in the order of appearance in a word-aligned sentence pair is used to construct component-based Markov Chains as translation probabilities. To compensate for deficiencies in coverage and reordering during decoding, successful hybrid phrase-based and N–gram-based SMT approaches have been developed [47, 48, 164]. A different application of components was considered in hierarchical SMT. In [153], components (minimum connected subgraphs) are convoluted for successfully computing more accurate phrase-level alignment probabilities.

Techniques that can directly assist the method discussed in Section 7.5 include smoothing translation probabilities [53, 132], segmenting sentences [13, 91, 158, 165], and forming robust component structures as in Chapter 5.

Our work can also provide the means for paraphrase generation with accompanying paraphrase probabilities. In this case, the equivalent $p(\tau|\mathrm{s})$ probabilities are computed as in [6].

7.7 Conclusions

The focus of this chapter has been the following research question:

RQ8 *Given an unseen phrase pair (s, t), how can one compute the translation probability $p(t|s)$ based on their most likely composition of building blocks?*

We showed how to compute translation probabilities based on structure provided by known alignments. This structure is the set of components that is formed in aligned sentence pairs. More accurately, the purpose was twofold: 1) To provide a theoretically sound framework for statistical bilingual compositionality, with components being the constituents of composition. 2) To derive translation probabilities from that framework for applications, and in particular for estimating translation probabilities of unseen phrase pairs.

For a given phrase pair we explained that there exists a dominant partition into components. Then, our key assertion was that the phrase pair obeys a statistical form of compositionality, with the constituents of composition being the components of that partition. After a series of approximations we formed a statistically elaborate relationship between translation probabilities of the phrase pair and translation probabilities of certain sub-phrase pairs.

Chapter 8

Conclusions

In this concluding chapter we summarize our answers to the research questions as stated in Chapter 1 and we provide pointers for future work (**FW**).

8.1 Looking back

RQ1 *How can one devise a mathematical framework that is affable to the consistency method? It should be minimal in construction but sufficient for accommodating bilingual segmentations as a generalization. If bilingual segmentations are taken into account, then how do they affect the set of extracted translation rules?* [Chapter 3]

First, it was shown that a word-aligned sentence pair has a graph representation as follows: Its source and target language words can be viewed as source and target type vertices respectively; word alignments play the role of edges that connect source and target type vertices. Such a graph is bipartite because no source-to-source nor target-to-target edges are assumed. Word alignments admit a natural partition for this graph: Each part is a connected component of the graph. We established that a phrase pair is a translation rule if and only if the following conditions hold: i) Its words respect the order of appearance in the sentence pair. ii) Its words are in one-to-one correspondence with the words of *a union of components of the bipartite graph*. Equivalently, a translation rule is formed by taking an arbitrary collection of components, extracting its words and then ordering them in a way so that the resulting phrase pair is a substring of the sentence pair.

Second, we have shown that the graph representation of a word-aligned sentence pair is just one possible configuration of a more general system: The one that allows consecutive words in a sentence to be connected via edges, for both sentences. Under this generalized system, it was shown that the same set of translation rules can be extracted in a different way: A phrase pair is a translation rule if and only if i) Its words respect the order of appearance in the sentence pair. ii) Its words are in one-to-one correspondence with the words of *a component of a configuration*. This implies that all translation rules can be recovered by collecting components (and components only, not arbitrary unions thereof) from all possible configurations.

RQ2 *How should one construct a method for computing probabilities of non–exchangeable random partitions?* [Chapter 3]

The probability of a partition of a finite set was viewed as a case of constrained, biased sampling without replacement. We derived a probability mass function that is close in construction to the Hyper-Dirichlet type I distribution. This was achieved by considering a partition as an outcome of all possible stochastic processes that lead to its formation. This assumption was coupled with a compact graph-based encoding of all possible partitions of a set.

RQ3 *Given a sentence in some language, identify what conditions a segmentation of the sentence should satisfy, in order for linear compositionality of meaning to hold. How can one define the segmentation that satisfies those conditions optimally?* [Chapter 4]

In order for the Principle of Compositionality to hold with the associated function being linear, we required the following condition for the segmentation: Substitution of its segments with their paraphrases yields new sentences which do not deviate much in meaning from the original sentence. Additionally, if segments are required to be minimal in size, then the segmentation that meets these two conditions is termed the natural segmentation of the sentence.

The above is an equivalent definition for the formal definition of the natural segmentation of a sentence that was presented, and stems from the definition of measure-theoretic entropy in dynamical systems.

RQ4 *Given the relationship between Shannon's entropy and metrics on lattices [138], elaborate on the mathematical framework of Pointwise Mutual Information (PMI). Is it possible to extend PMI within this framework for simulating natural segmentations?* [Chapter 4]

The method that was found to simulate natural segmentations dealt with estimating costs for perturbing a segmentation into another. For a particular sentence, the set of all its segmentations together with the operation of 'refinement' (splitting of a segment into two new segments) forms a partially ordered set (poset). Choosing a metric on that poset appropriately, resulted in the formation of cost functions for segments and, consequently, segmentations of that sentence. The optimal segmentation was calculated as the one that is the least expensive to be perturbed into. The development of this method revealed the previously unseen theoretical link for PMI: It was shown that PMI is a metric on the said poset.

RQ5 *Given a word-aligned sentence pair, identify what conditions a bilingual segmentation of the pair should satisfy in order to form a bilingual natural segmentation. How can one define the bilingual segmentation that satisfies those conditions optimally?* [Chapter 5]

This was achieved by identifying two necessary conditions: 1) Segments in both source and target language sentences abide to natural segmentation. 2) Source and target language segments are synchronized with each other via word alignments.

Condition 1 is covered by the conditions of RQ3. Condition 2 was identified as the necessary condition that permits the generalization to the bilingual setting. It refers to purely structural properties of the graph representation of a given word-aligned bilingual segmentation, as discussed in RQ1. Its components were assessed according to how well source and target segments are synchronized in the word-aligned bilingual segmentation. Based on the notion of connecting spanning subgraph (CSSG) from graph theory, we introduced a measure that assesses structural robustness of components. This measure counts a particular type of CSSGs of a component, namely the connected spanning subgraphs of the component from which only translation edges are allowed to be deleted. The appropriately normalized value of this quantity essentially tells how difficult it is to perturb the component from its connected state into a disconnected state.

Conditions 1 and 2 gave rise to surface and structural robustness quantitative criteria respectively. The former can be viewed as 'Precision' and the latter as 'Recall'. Thus, a bilingual natural segmentation was defined as a bilingual segmentation that provides a balance between Precision and Recall, i.e., the harmonic mean of surface and structural measures.

RQ6 *What is the effect of bilingual natural segmentations on SMT?* [Chapter 5]

Phrase-tables were formed by extracting translation rules from components only (and not unions thereof) of configurations in the vicinity of the bilingual natural segmentations. Translation probabilities were computed in a similar way to the standard method, but counts were collected from this new set of translation rules; all other features were the same as with the baseline (Moses). Experiments were performed on German–English and Czech–English language pairs with strong baselines and language models. Phrase-table sizes were 90% smaller and, most importantly, translation quality was shown to be comparable with the baseline. The latter is a fruitful result as it reveals the qualitative characteristics of the phrase pairs that are useful to SMT.

RQ7 *How should one extend the work of Kok and Brockett [87] in order to identify less noisy pairs of paraphrases and to develop a method that constructs artificial co-occurrence counts for these pairs?* [Chapter 6]

Our method relied on forming a collection of sub-phrase-tables which is mainly constructed from connected components of the phrase-table's graph representation. The connectivity of each sub-phrase-table was exploited for the purpose of creating separate source phrase and target phrase clusters. Each cluster inherits a new weighted graph structure from its corresponding sub-phrase-table. The commute time between two vertices was then employed for finding the degree of similarity between any two phrases in a cluster. An important distinction of this method from previous work is that a pair of phrases may appear in multiple clusters across multiple sub-phrase-tables. This allowed us to compute their expected similarity, thus significantly reducing the effects of noise. The involved probability distribution was obtained through a novel heuristic iterative method. This method also produced the conversion of expected commute time distances between a pair of phrases into co-occurrence counts of such a pair. These counts were interpreted as artificial co-occurrence counts between a pair of source (target) phrases in hypothetical source (target) language-to-paraphrased source (target) language aligned corpora.

RQ8 *Given an unseen phrase pair (s, t), how can one compute the translation probability $p(t|s)$ based on their most likely composition of building blocks?* [Chapter 7]

Our key assertion was that the (any) phrase pair obeys a statistical form of linear compositionality. The constituents of composition are the most likely building blocks that make up the phrase pair. After a series of approximations we formed a statistically elaborate relationship between translation probabilities of the phrase pair and translation probabilities of certain sub-phrase pairs. Experiments showed that this approach was superior to simple heuristics.

8.2 Looking forward

FW1 In Section 3.4 we showed that the phrase-table in SMT is constructed from the powerset of components emerging from word-aligned sentence pairs in the training data. The experiments of Section 5.5 showed that much fewer translation rules are actually needed. This effective set of translation rules consists mainly of components emerging from bilingual segmentations that are in the vicinity of the bilingual natural segmentations of the training data. The boundary of this domain was set heuristically (parameter N in Section 5.5). Can we determine this boundary algorithmically?

FW2 Investigate the combinatorial properties of graphs induced by segmentations in Section 3.6.2: If $G_n = (V_n, E_n)$ denotes such a graph that corresponds to a sentence with n words, then how do $|V_n|$, $|E_n|$, chromatic number, maximal cliques, etc. depend on n?

FW3 The formula for non-exchangeable random partitions that was introduced in Section 3.6.2 is impractical for long sentences. Develop sampling techniques that make such computations possible.

FW4 Are there any applications of the varied n-gram language models of Section 4.2 to SMT? In particular, can this method help the ranking process of candidate translations during decoding?

FW5 Is it possible to formalize the segments-segmentations relationship of Section 4.3.3 via the Moebius Inversion Theorem?

FW6 The methods that were introduced in Chapter 4 are certainly not restricted to applications in SMT. How can refinement-based phrase segmentations assist other branches of computational linguistics?

FW7 Integrate the notion of natural segmentation (Section 4.5.1) with Hodges' canonical composition in formal semantics [68] and with measure theoretic entropy in dynamical systems [77, 120]. This is perhaps the most interesting and challenging task that arises from this thesis, with several potential applications in computational linguistics.

FW8 The method for harvesting paraphrases (Chapter 6) is admittedly too complicated to be used as a whole. Nonetheless, the algorithms described terein can be used inpependently for applications. Of particular interest is the algorithm that converts distances into probabilities (Section 6.5) and should be explored further.

FW9 Chapter 7 provides a clear distinction between strings, sets of words, and ordering of words in strings, all with associated random variables. The notions that result in method `Permute` also encompass both basic heuristics (average and product of the most likely bilingual chunks). Does this method provide a stepping stone for ad-hoc translations?

Appendix A

Maximum Likelihood Estimation of Language Model Probabilities

Let a corpus be the sequence of words $w_1, ..., w_N$. The likelihood of the corpus is given by

$$p(w_1^N) = \prod_{i=1}^{N} p(w_i|\, w_1^{i-1}) \approx \prod_{i=1}^{N} p_n(w_i|\, w_{i-n+1}^{i-1}), \tag{A.1}$$

where the approximation happens because we consider fixed memory of $n-1$ preceding words, for each word in the corpus. The goal is to find the conditional probabilities of the n-gram LM. The log-likelihood of (A.1) is

$$\ell_n = \sum_{i=1}^{N} \log p_n(w_i|\, w_{i-n+1}^{i-1}), \tag{A.2}$$

which should be maximized subject to the constraints

$$\sum_v p_n(v|y) = 1, \quad \text{for all sequences } y. \tag{A.3}$$

The objective function is thus

$$F = \ell_n - \sum_y \lambda_y \sum_v p_n(v|y), \tag{A.4}$$

where λ_y's are the Lagrange multipliers. Extrema are given by the roots of $\partial F/\partial p_n(w|h) = 0$, or

$$\sum_{i=1}^{N} \frac{\delta(w, w_i)\, \delta(h, w_{i-n+1}^{i-1})}{p_n(w|h)} \;-\; \sum_y \lambda_y\, \delta(h, y) \;=\; 0, \tag{A.5}$$

where $\delta(a, b) = 1$, if $a = b$ and $\delta(a, b) = 0$, if $a \neq b$. The solution of (A.5) is given by

$$p_n(w|h) = \frac{\sum_{i=1}^{N} \delta(w, w_i)\, \delta(h, w_{i-n+1}^{i-1})}{\sum_y \lambda_y\, \delta(h, y)}, \tag{A.6}$$

where the numerator counts how many times the sequence hw has been observed in the corpus, or count(hw) for short. From the constraints we must have $\sum_v p_n(v|h) = 1$, or

$$\begin{aligned}\sum_v \text{count}(hv) &= \sum_y \lambda_y \, \delta(h,y) \quad \Leftrightarrow \\ \text{count}(h) &= \sum_y \lambda_y \, \delta(h,y). \qquad \text{(A.7)}\end{aligned}$$

But we also have count$(h) = \sum_y \delta(h,y)$, so that we can set $\lambda_y = 1$, for all sequences y. Thus, (A.6) becomes

$$p_n(w|h) = \frac{\text{count}(hw)}{\text{count}(h)}, \quad \text{for any sequence } hw. \qquad \text{(A.8)}$$

It is also trivial to show that $\partial^2 F / \partial p_n(w|h)^2$ is negative when evaluated at (A.8), so that (A.8) indeed maximize the likelihood of the corpus.

Appendix B

Combinatorial Optimization and the Cross-Entropy Method

Let X be a finite set and let $F : X \to \mathbb{R}$ denote a deterministic function that evaluates the performance of X's elements. We assume that X is large enough so that the problem

$$\gamma^* = F(x^*) = \max_{x \in X} F(x) \tag{B.1}$$

is difficult to solve, even if $F(x)$ is computed in linear time, for any x. Typically, each $x \in X$ is a vector representing a configuration of a system, so that X is the set of all possible configurations.

The CE method is an efficient way for obtaining the best performing configuration x^*. It requires the conversion of (B.1) into a stochastic problem which gives rise to an iterative algorithm. At each iteration t of the algorithm, a small random sample O^t is drawn from a probability mass function (pmf) with parameter θ that is associated with X. Thus, all instances of the random set O^t are elements of X, but O^t is such that $|O^t|/|X| \ll 1$. The aim is to appropriately update θ at every iteration so that, as t increases, the instances of O^t tend to focus around the region of X that includes x^*. After enough iterations, say at $t = t_\infty$, θ should be a particular kind of pmf, namely a Dirac measure: Its mass is entirely allocated to a single point and, for our purposes, this point should ideally be the argument that maximizes F. It follows that all instances of the random sample at t_∞ are x^*, which would provide the solution to (B.1).

In more detail, the stochastic problem that is associated with (B.1) requires each $x \in X$ to be drawn with probability $f(x;\theta)$, where θ is a possibly multidimensional parameter. The pmf $f(\cdot;\theta)$ is assumed known (e.g. via the Maximum Entropy method) and is a member of the family of pmfs $\{f(\cdot;\lambda)\}_\lambda$. The exact value of θ does not play any role in the CE method for combinatorial optimization.

For a given value of θ the quantity of interest is

$$\ell(\gamma) = \mathbb{P}_\theta(F(y) \geq \gamma), \quad \text{with } y \sim f(\cdot;\theta), \tag{B.2}$$

so that $\ell(\gamma)$ is the probability that configurations drawn from $f(\cdot;\theta)$ perform at least γ. If $\gamma > \gamma^*$, then we have $\ell(\gamma) = 0$. This is because F is assumed deterministic and no configuration performs better than γ^*, regardless of the details of $f(\cdot;\theta)$. On the other hand, no immediate conclusions can be drawn for $\ell(\gamma)$ when $\gamma \leq \gamma^*$. The aim is to find θ such that $\tilde{\ell}(\gamma^*) = 1$, for a renormalized version $\tilde{\ell}$ of ℓ.

For a given value of $\gamma \in \mathbb{R}$, let

$$I_{\{F(x)\geq\gamma\}} = \begin{cases} 1, & \text{if } F(x) \geq \gamma \\ \\ 0, & \text{otherwise,} \end{cases} \tag{B.3}$$

denote the indicator function on X. Then (B.2) can be rewritten as

$$\ell(\gamma) = \sum_{x\in X} I_{\{F(x)\geq\gamma\}}\ f(x;\theta), \tag{B.4}$$

so that $\ell(\gamma)$ is the expected value of $I_{\{F(y)\geq\gamma\}}$ under $f(\cdot;\theta)$. An unbiased estimator of (B.4) is

$$\hat{\ell}(\gamma) = \frac{1}{|O|}\sum_{x\in O} I_{\{F(x)\geq\gamma\}}, \qquad \text{with } O \sim f(\cdot;\theta). \tag{B.5}$$

In words, O is a random sample, i.e., a set of random instances, and each such instance is drawn from $f(\cdot;\theta)$. Quantity $\hat{\ell}(\gamma)$ measures how many of those instances perform at least γ on average. (B.5) is not directly useful for our purposes. Instead, we consider a new pmf h for X and rewrite (B.4) as

$$\ell(\gamma) = \sum_{x\in X} I_{\{F(x)\geq\gamma\}}\ \frac{f(x;\theta)}{h(x)}\ h(x), \tag{B.6}$$

so that $\ell(\gamma)$ is the expected value of $I_{\{F(y)\geq\gamma\}}\ f(y;\theta)/h(y)$ under h. Similarly, an unbiased estimator of (B.6) is

$$\hat{\ell}(\gamma) = \frac{1}{|O|}\sum_{x\in O} I_{\{F(x)\geq\gamma\}}\ \frac{f(x;\theta)}{h(x)}, \qquad \text{with } O \sim h. \tag{B.7}$$

Pmf h is called the importance sampling and operates as a surrogate pmf; it will be constructed in a way so that

$$h(x) = \begin{cases} 1, & \text{if } x = x^* \\ \\ 0, & \text{otherwise,} \end{cases} \tag{B.8}$$

which would then imply $\hat{\ell}(\gamma) = 0$, whenever $\gamma \neq \gamma^*$.

It is known [76] that the most appropriate choice for h is

$$h^*(x) = \frac{I_{\{F(x)\geq\gamma\}}\ f(x;\theta)}{\sum_{x'\in X} I_{\{F(x')\geq\gamma\}}\ f(x';\theta)} = \frac{I_{\{F(x)\geq\gamma\}}\ f(x;\theta)}{\ell(\gamma)}, \tag{B.9}$$

which is seemingly useless as it depends on $\ell(\gamma)$. For simplicity we require h to be a pmf from the family of pmfs $\{f(\cdot;\lambda)\}_\lambda$, i.e., $h = f(\cdot;\lambda^*)$, for some λ^* that needs to be determined. The key step of the CE method requires λ^* to be such that the Kullback–Leibler divergence of $f(\cdot;\lambda^*)$ from h^* is minimal. In other words, we require

$$\begin{aligned}\lambda^* &= \operatorname*{argmin}_\lambda D(h^*||f(\cdot;\lambda)) = \operatorname*{argmin}_\lambda \sum_{x\in X} h^*(x)\ln\frac{h^*(x)}{f(x;\lambda)}\\ &= \operatorname*{argmin}_\lambda \left\{\sum_{x\in X} h^*(x)\ln h^*(x) - \sum_{x\in X} h^*(x)\ln f(x;\lambda)\right\}\\ &= \arg\max_\lambda \sum_{x\in X} h^*(x)\ln f(x;\lambda). \end{aligned} \tag{B.10}$$

Inserting (B.9) in (B.10) results in

$$\lambda^* = \arg\max_\lambda \sum_{x\in X} \frac{I_{\{F(x)\geq\gamma\}}\ f(x;\theta)}{\ell(\gamma)} \ln f(x;\lambda). \tag{B.11}$$

According to (B.4) $\ell(\gamma)$ is independent of λ and (B.11) becomes

$$\lambda^* = \arg\max_\lambda \sum_{x\in X} I_{\{F(x)\geq\gamma\}}\ f(x;\theta)\ln f(x;\lambda), \tag{B.12}$$

which can be approximated as

$$\lambda^* = \arg\max_\lambda \frac{1}{|A|}\sum_{x\in A} I_{\{F(x)\geq\gamma\}}\ \ln f(x;\lambda), \quad \text{with } A\sim f(\cdot;\theta). \tag{B.13}$$

In many cases it is even possible to find λ^* analytically and (B.7) becomes

$$\hat{\ell}(\gamma) = \frac{1}{|O|}\sum_{x\in O} I_{\{F(x)\geq\gamma\}}\ \frac{f(x;\theta)}{f(x;\lambda^*)}, \quad \text{with } O\sim f(\cdot;\lambda^*). \tag{B.14}$$

If the true θ of $f(\cdot;\theta)$ is known, then the purpose of constructing $f(\cdot;\lambda^*)$ is for reallocating most probability mass around the subset of X of our desire. Since (B.14) is the same as (B.5), no direct gains can be earned from (B.14). The quantity that characterizes how $f(\cdot;\lambda^*)$ reshapes the probability with which a sample performs at least γ on average is given by

$$\tilde{\ell}(\gamma) = \frac{1}{|O|}\sum_{x\in O} I_{\{F(x)\geq\gamma\}}, \quad \text{with } O\sim f(\cdot;\lambda^*), \tag{B.15}$$

which concludes the involvement of importance sampling techniques in the CE method. Clearly, if $f(\cdot;\lambda^*)$ is of the form (B.8), then $\tilde{\ell}(\gamma^*) = 1$. All equations that involve pmfs in this section are of no practical use if a single sample for each of O and A is drawn.

The algorithm of the CE method requires an iterative application of (B.13) and (B.15) in order to construct the surrogate pmf of (B.8).

In particular, for the iterative algorithm of the CE method, quantities γ, O and θ are denoted at each iteration t by γ^t, O^t and θ^t respectively. Let N be a positive integer with $N/|X| \ll 1$ and set $\rho = \lceil 1\% N \rceil = \lceil N/100 \rceil$. The algorithm is as follows.

1. Set some arbitrary value θ^0, such that the pmf of the associated stochastic problem to (B.1) is $f(\cdot;\theta^0)$. Also, set $t = 1$, the level counter of the iterative process.

2. Draw a random sample $O^t \sim f(\cdot;\theta^{t-1})$ with $|O^t| = N$. Compute scores $F(x_1)$,...,$F(x_N)$, where $x_i \in O^t$, $i = 1, ..., N$. Order them descendingly, i.e., find permutation π of $\{1, ..., N\}$ such that $F(x_{\pi(1)}) \geq ... \geq F(x_{\pi(N)})$.

3. Focus on the best performing subsample of O^t: Compute performance threshold $\gamma^t = F(x_{\pi(\rho)})$.

4. Compute
$$\theta^t = \arg\max_{\lambda} \frac{1}{N} \sum_{x \in O^t} I_{\{F(x) \geq \gamma^t\}} \; \ln f(x;\lambda), \tag{B.16}$$
where O^t is the same sample as in Step 2.

5. If for some $t \geq d$, say $d = 5$, we have $\gamma^t = \gamma^{t-1} = ... = \gamma^{t-d}$, then stop. Else, set $t := t + 1$ and go to Step 2.

Observe that probability $\tilde{\ell}(\gamma^t)$ of (B.15) is not computed explicitly. At every iteration, integer ρ provides the cut-off for the best performing subsample of O^t, which gives rise to threshold γ^t. Then, from (B.16) a new pmf $f(\cdot;\theta^t)$ is constructed which assigns higher probability than $f(\cdot;\theta^{t-1})$ to configurations that perform in the vicinity of γ^t. Typically $\gamma^t \geq \gamma^{t-1}$ and after enough iterations, γ^t gets close to the highest performance γ^*. Once $\gamma^t = \gamma^*$, it is then (ideally) a matter of a few more iterations, say at $t = t_\infty$, until the Dirac measure $f(x^*;\theta^{t_\infty}) = 1$ is constructed. This also implies that $O^{t_\infty} = \{x^*, ..., x^*\}$ and $\tilde{\ell}(\gamma^*) = 1$.

Appendix C

A Statistically Elaborate Relationship Between a Phrase Pair and Its Structure

Given phrase pair (s, t) with structure $g(s, t) = \{(s_1, t_1), ..., (s_k, t_k)\}$, we sketch the proof for

$$p(t|s) = \sum_{c=1}^{k} p(t \setminus t_c | s \setminus s_c) p(s_c, t_c | s) \tag{C.1}$$

$$= \sum_{\pi} \prod_{c=1}^{k} p(s_{\pi(c)}, t_{\pi(c)} | s \setminus \bigcup_{i=0}^{c-1} s_{\pi(c)}), \tag{C.2}$$

where the sum in (C.2) is over all permutations of $\{0, 1, ..., k\}$ with $\pi(0) = 0$.

Choose any of the summands of (C.1), say

$$p(t \setminus t_{c_1} | s \setminus s_{c_1}) \times p(s_{c_1}, t_{c_1} | s), \tag{C.3}$$

with $c_1 \in \{1, ..., k\}$. But $p(t \setminus t_{c_1} | s \setminus s_{c_1})$ is a source string-to-target string probability for phrase pair $(s \setminus s_{c_1}, t \setminus t_{c_1})$ with structure $g_1(s, t) = g(s, t) \setminus \{(s_{c_1}, t_{c_1})\}$, and can thus be written in the form of (C.1) as well. This yields a new sum from which we can choose a new degenerate summand indexed by, say c_2, and whose corresponding phrase pair has structure $g_2(s, t) = g_1(s, t) \setminus \{(s_{c_2}, t_{c_2})\}$. We repeat this process until there are no more components to delete; the base case of this iterative process is

$$p(t \setminus \{t_{c_1}, ..., t_{c_k}\} \mid s \setminus \{s_{c_1}, ..., s_{c_k}\}) \times p(s_{c_k}, t_{c_k} \mid s \setminus \{s_{c_1}, ..., s_{c_{k-1}}\}), \tag{C.4}$$

for which we have $c_i \neq c_j$, $\forall i \neq j$. Since there are k such indices it must be that the sequence $c_1, ..., c_k$ is a permutation of $\{1, ..., k\}$. Also, the source string-to-target string probability of the base case becomes $p(\emptyset|\emptyset) = 1$. This sequence of choices that we made in the iterative process yields the product

$$\prod_{c=1}^{k} p(s_{\pi(c)}, t_{\pi(c)} | s \setminus \bigcup_{i=0}^{c-1} s_{\pi(c)}),$$

and by considering all possible choices we get (C.2) as required.

Bibliography

[1] Aggarwal, C. C. and Zhai, C. (2012). A Survey of Text Clustering Algorithms. In *Mining Text Data*, pages 77–128, Springer US.

[2] Aldous, D. and Fill, J. (2002). *Reversible Markov Chains and Random Walks on Graphs*. `http://www.stat.berkeley.edu/~aldous/RWG/book.html`

[3] Androutsopoulos, I. and Malakasiotis, P. (2010). A Survey of Paraphrasing and Textual Entailment Methods. *Journal of Artificial Intelligence Research*, Vol. 38, pages 135–187.

[4] Ayan, N. F., Dorr, B. J., and Monz, C. (2005). Neuralign: Combining Word Alignments Using Neural Networks. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 65–72.

[5] Bach, E. (1989). *Informal Lectures on Formal Semantics*. State University of New York Press, Albany, NY.

[6] Bannard, C. and Callison-Burch, C. (2005). Paraphrasing with Bilingual Parallel Corpora. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 597–604.

[7] Bansal, M., Quirk, C., and Moore, R. C. (2011). Gappy Phrasal Alignment by Agreement. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Vol. 1, pages 1308–1317.

[8] Baroni, M., Bernardi, R., and Zamparelli, R. (2014). Frege in Space: A Program for Compositional Distributional Semantics. *Linguistic Issues in Language Technology*, Vol. 9, pages 5–110.

[9] Barrat, A., Barthélemy, M., and Vespignani, A. (2004). Modeling the Evolution of Weighted Networks. *Physical Review Letters*, Vol. 70(6), Article ID 066149.

[10] Berestycki, N. (2009). Recent Progress in Coalescent Theory. *Ensaios Matematicos*, Vol. 16(1).

[11] Birch, A., Callison-Burch, C., Osborne, M., and Koehn, P. (2006). Constraining the Phrase-based, Joint Probability Statistical Translation Model. In *Proceedings of the Workshop on Statistical Machine Translation*, pages 154–157.

[12] Birkhoff, G. (1948). *Lattice Theory*. Vol. 25, New York: American Mathematical Society.

[13] Blackwood, G., de Gispert, A., and Byrne, W. (2008). Phrasal Segmentation Models for Statistical Machine Translation. In *Coling 2008: Companion volume: Posters and Demonstrations*, pages 19–22.

[14] Bojar, O., Buck, C., Callison-Burch, C., Federmann, C., Haddow, B., Koehn, P., Monz, C., Post, M., Soricut, R., and Specia, L. (2013). Findings of the 2013 Workshop on Statistical Machine Translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 1–44.

[15] Bojar, O., Chatterjee, R., Federmann, C., Haddow, B., Huck, M., Hokamp, C., Koehn, P., Logacheva, V., Monz, C., Negri, N., Post, M., Scarton, C., Specia, L., and Turchi, M. (2015). Findings of the 2015 Workshop on Statistical Machine Translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 1–46.

[16] Bojar, O. and Tamchyna, A. (2015). CUNI in WMT15: Chimera Strikes Again. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 79–83.

[17] Boley, D., Ranjan, G., and Zhang, Z. L. (2011). Commute Times for a Directed Graph using an Asymmetric Laplacian. *Linear Algebra and its Applications*, Vol. 435(2), pages 224–242.

[18] Brants, T., Popat, A. C., Xu, P., Och, F. J., and Dean, J. (2007). Large Language Models in Machine Translation. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 858–867.

[19] Brown, P. F., Della Pietra, S. A., Della Pietra, V. J., and Mercer, R. L. (1993). The Mathematics of Statistical Machine Translation. *Computational Linguistics*, Vol. 19(2), pages 263–312.

[20] Burkett, D., Blitzer, J., and Klein, D. (2010). Joint Parsing and Alignment with Weakly Synchronized Grammars. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 127–135.

[21] Callison-Burch, C., Koehn, P., and Osborne, M. (2006). Improved Statistical Machine Translation Using Paraphrases. In *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 17–24.

[22] Callison-Burch, C., Fordyce, C., Koehn, P., Monz, C., and Schroeder, J. (2007). (Meta-) Evaluation of Machine Translation. In *Proceedings of the Workshop on Statistical Machine Translation*, pages 136–158.

[23] Callison-Burch, C. (2008). Syntactic Constraints on Paraphrases Extracted from

Parallel Corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 196–205.

[24] Cettolo, M., Federico, M., and Bertoldi, N. (2010). Mining Parallel Fragments from Comparable Texts. In International Workshop on Spoken Language Translation, pages 227–234.

[25] Chen, S. F. and Goodman, J. (1998). An Empirical Study of Smoothing Techniques for Language Modeling. *Centre for Research in Computing Technology, Harvard University, Cambridge, Massachusetts, USA, Tech. Rep. TR-10-98.*

[26] Cherry, C. and Lin, D. (2007). Inversion Transduction Grammar for Joint Phrasal Translation Modeling. In *Proceedings of the NAACL-HLT 2007/AMTA Workshop on Syntax and Structure in Statistical Translation*, pages 17–24.

[27] Chiang, D. (2007). Hierarchical Phrase-based Translation. *Computational Linguistics*, Vol. 33(2), pages 201–228.

[28] Coecke, B., Sadrzadeh, M., and Clark, S. (2011). Mathematical Foundations for a Compositional Distributional Model of Meaning. *Linguistic Analysis: A Festschrift for Joachim Lambek*, Vol. 36(1-4), pages 345–384.

[29] Colbourn, C. J. (1987). *The Combinatorics of Network Reliability*. Oxford University Press.

[30] Collins, M., Koehn, P., and Kučerová, I. (2005). Clause Restructuring for Statistical Machine Translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 531–540.

[31] Conway, H. J. and Guy, R. K. (1996). *The Book of Numbers*. Copernicus.

[32] Cover, T. M. and Thomas, J. A. (2012). *Elements of Information Theory*. John Wiley & Sons.

[33] Crego, J. M. and Yvon, F. (2009). Gappy Translation Units Under Left-to-Right SMT Decoding. In *Proceedings of the 13th Annual Conference of the European Association for Machine Translation*, pages 66–73.

[34] De Boer, P. T., Kroese, D. P., Mannor, S., and Rubinstein, R. Y. (2005). A Tutorial on the Cross-Entropy Method. *Annals of Operations Research*, Vol. 134(1), pages 19–67.

[35] de Gispert, A., Iglesias, G., and Byrne, B. (2015). Fast and Accurate Preordering for SMT using Neural Networks. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1012–1017.

[36] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society*, Series B (Methodological), Vol. 39(1), pages 1–38.

[37] DeNero, J., Gillick, D., Zhang, J., and Klein, D. (2006). Why Generative Phrase Models Underperform Surface Heuristics. In *Proceedings of the Workshop on Statistical Machine Translation*, pages 31–38.

[38] DeNero, J. and Klein, D. (2007). Tailoring Word Alignments to Syntactic Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, Vol. 45(1), pages 17–24.

[39] DeNero, J. and Klein, D. (2008). The Complexity of Phrase Alignment Models. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, pages 25–28.

[40] DeNero, J. and Klein, D. (2010). Discriminative Modeling of Extraction Sets for Machine Translation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1453–1463.

[41] Deng, Y. and Byrne, W. (2008). HMM Word and Phrase Alignment for Statistical Machine Translation. *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 16(3), 494–507.

[42] Deng, Y. and Gao, Y. (2007). Guiding Statistical Word Alignment Models with Prior Knowledge. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, Vol. 45(1), pages 1–8.

[43] Dennis III, S. Y. (1991). On the Hyper-Dirichlet Type 1 and Hyper-Liouville Distributions. *Communications in Statistics - Theory and Methods*, Vol. 20(12), pages 4069–4081.

[44] Dhillon, I., Mallela, S., and Modha, D. (2003). Information-Theoretic Coclustering. In *Proceedings of the ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 89–98.

[45] Dhillon, I. and Guan, Y. (2003). Information Theoretic Clustering of Sparse Co-Occurrence Data. In *Third IEEE International Conference on Data Mining*, pages 517–521.

[46] Dolan, W., Quirk, C., and Brockett, C. (2004). Unsupervised Construction of Large Paraphrase Corpora: Exploiting Massively Parallel News Sources. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 350–356.

[47] Durrani, N., Fraser, A., Schmid, H., Hoang, H., and Koehn, P. (2013). Can Markov Models Over Minimal Translation Units Help Phrase-Based SMT? In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 399–405.

[48] Durrani, N., Schmid, H., and Fraser, A. (2013). Model with Minimal Translation Units, But Decode With Phrases. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (HLT-NAACL)*, pages 1–11.

[49] Dyer, C., Clark, J., Lavie, A., and Smith, N. A. (2011). Unsupervised Word Alignment with Arbitrary Features. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Vol. 1, pages 409–419.

[50] Feller, W. (1971). *An Introduction to Probability Theory and its Applications, Volume II.* John Wiley, New York.

[51] Ferguson, T. S. (1973). A Bayesian Analysis of Some Nonparametric Problems. *Annals of Statistics*, Vol. 1(2), pages 209–230.

[52] Ferrer, J. A. and Juan, A. (2009). A Phrase-based Hidden Semi-Markov Approach

to Machine Translation. In *Proceedings of the 13th Annual Conference of the European Association for Machine Translation*, pages 168–175.

[53] Foster, G., Kuhn, R., and Johnson, H. (2006). Phrasetable Smoothing for Statistical Machine Translation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*, pages 53–61.

[54] Galley, M. and Manning, C. D. (2010). Accurate Non-Hierarchical Phrase-based Translation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 966–974.

[55] Ganitkevitch, J., Callison-Burch, C., Napoles, C., and Van Durme, B. (2011). Learning Sentential Paraphrases from Bilingual Parallel Corpora for Text-to-Text Generation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1168–1179.

[56] Gimpel, K. and Smith, N. A. (2011). Generative Models of Monolingual and Bilingual Gappy Patterns. In *Proceedings of the Workshop on Statistical Machine Translation*, pages 512–522.

[57] Gnedin, A., Haulk, C., and Pitman, J. (2009). Characterizations of Exchangeable Partitions and Random Discrete Distributions by Deletion Properties. *arXiv preprint arXiv:0909.3642*.

[58] Goodman, J. T. (2001). A Bit of Progress in Language Modeling Extended Version. *Machine Learning and Applied Statistics Group Microsoft Research. Technical Report, MSR-TR-2001-72*.

[59] Green, S., Galley, M., and Manning, C. D. (2010). Improved Models of Distortion Cost for Statistical Machine Translation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 867–875.

[60] Griffiths, T. L. and Ghahramani, Z. (2005). Infinite Latent Feature Models and the Indian Buffet Process. In *Advances in Neural Information Processing Systems*, pages 475–482.

[61] Grinstead, C. M. and Snell, J. L. (2006). *Introduction to Probability*. American Mathematical Society.

[62] Gusfield, D. (1997). *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology*. Cambridge University Press, Cambridge, UK.

[63] Guo, J. (1997). Critical Tokenization and its Properties. *Computational Linguistics*, Vol. 23(4), pages 569–596.

[64] Guo, J., Xu, G., Li, H., and Cheng, X. (2008). A Unified and Discriminative Model for Query Refinement. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 379–386.

[65] Haffari, G. and Teh, Y. W. (2009). Hierarchical Dirichlet Trees for Information Retrieval. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 173–181.

[66] Hagen, M., Potthast, M., Stein, B., and Bräutigam, C. (2011). Query Segmentation Revisited. In *Proceedings of the 20th International Conference on World Wide Web*, pages 97–106.

[67] Harary, F. (1969). *Graph Theory*. Addison–Wesley, Reading, MA.

[68] Hodges, W. (2001). Formal Features of Compositionality. *Journal of Logic, Language, and Information*, Vol. 10(1), Special Issue on Compositionality (Winter, 2001), pages 7–28

[69] Huang, X., Alleva, F., Hon, H. W., Hwang, M. Y., Lee, K. F., and Rosenfeld, R. (1993). The SPHINX-II Speech Recognition System: An Overview. *Computer, Speech, and Language*, Vol. 7(2), pages 137–148.

[70] Huck, M., Scharwächter, E., and Ney, H. (2013). Source-Side Discontinuous Phrases for Machine Translation: A Comparative Study on Phrase Extraction and Search. *The Prague Bulletin of Mathematical Linguistics*, Vol. 99, pages 17–38.

[71] Hunter, J. J. (2000). A Survey of Generalized Inverses and their Use in Stochastic Modelling. *Research Letters in the Information and Mathematical Sciences*, Vol. 1, pages 25–36.

[72] Ittycheriah, A. and Roukos, S. (2005). A Maximum Entropy Word Aligner for Arabic-English Machine Translation. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 89–96.

[73] Jean, S., Firat, O., Cho, K., Memisevic, R., and Bengio, Y. (2015). Montreal Neural Machine Translation Systems for WMT15. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 134–140.

[74] Johnson, J. H., Martin, J., Foster, G., and Kuhn, R. (2007). Improving Translation Quality by Discarding Most of the Phrasetable. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 967–975.

[75] Jones, R., Rey, B., Madani, O., and Greiner, W. (2006). Generating Query Substitutions. In *Proceedings of the 15th International Conference on World Wide Web*, pages 387–396.

[76] Kahn, H. and Marshall, A. (1953). Methods of Reducing Sample Size in Monte Carlo Computations. *Journal of the Operations Research Society of America*, Vol. 1(5), pages 263–278.

[77] Katok, A. (1980). Lyapunov Exponents, Entropy and Periodic Orbits for Diffeomorphisms. *Publications Mathématiques de l'IHÉS*, Vol. 51, pages 137–173.

[78] Katz, J. J. and Fodor, J. A. (1963). The Structure of a Semantic Theory *Language*, Vol. 39(2), pages 170–210.

[79] Kneser, R. and Ney, H. (1995). Improved Backing-Off for M-Gram Language Modeling. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 1, pages 181–184.

[80] Kneser, R. (1996). Statistical Language Modeling Using a Variable Context

Length. In *Proceedings of the Fourth International Conference on Spoken Language Processing*, Vol. 1, pages 494–497.

[81] Knight, K. (1999). Decoding Complexity in Word-Replacement Translation Models. *Computational Linguistics*, Vol. 25(4), pages 607–615.

[82] Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical Phrase-based Translation. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (HLT-NAACL)*, pages 48–54.

[83] Koehn, P. (2005). Europarl: A Parallel Corpus for Statistical Machine Translation. In *MT Summit*, Vol. 5, pages 79–86.

[84] Koehn, P., Axelrod, A., Birch, A., Callison-Burch, C., Osborne, M., Talbot, D., and White, M. (2005). Edinburgh System Description for the 2005 IWSLT Speech Translation Evaluation. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT-05)*, pages 68–75.

[85] Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions (ACL '07).*

[86] Koehn, P. (2009). *Statistical Machine Translation.* Cambridge University Press, Cambridge, UK.

[87] Kok, S. and Brockett, C. (2010). Hitting the Right Paraphrases in Good Time. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 145–153.

[88] Kroch, A. (1989). Reflexes of Grammar in Patterns of Language Change. *Language Variation and Change*, Vol. 1(3), pages 199–244.

[89] Kuhn, R., Chen, B., Foster, G., and Stratford, E. (2010). Phrase Clustering for Smoothing TM Probabilities: or, how to Extract Paraphrases from Phrase Tables. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, Vol. 2, pages 608–616.

[90] Lappin, S. and Zadrozny, W. (2000). Compositionality, Synonymy, and the Systematic Representation of Meaning. *arXiv preprint cs/0001006.*

[91] Lee, H.-G., Lee, J.-Y., Kim, M.-J., Rim, H.-C., Shin, J.-H., and Hwang, Y.-S. (2011). Phrase Segmentation Model Using Collocation and Translational Entropy. In *Proceedings of MT Summit XIII.*

[92] Lerner, U. and Petrov, S. (2013). Source-Side Classifier Preordering for Machine Translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 513–523.

[93] Liu, Y., Liu, Q., and Lin, S. (2010). Discriminative Word Alignment by Linear Modeling. *Computational Linguistics*, Vol. 36(3), pages 303–339.

[94] Madnani, N. and Dorr, B. (2010). Generating Phrasal and Sentential Paraphrases:

A Survey of Data-Driven Methods. *Computational Linguistics*, Vol. 36(3), pages 341–388.

[95] Marcu, D. and Wong, W. (2002). A Phrase-based, Joint Probability Model for Statistical Machine Translation. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing*, Vol. 10, pages 133–139.

[96] Marie, B., Allauzen, A., Burlot, F., Do, Q.-K., Ive, J., Knyazeva, E., Labeau, M., Lavergne, T., Löser, K., Pécheux, N., and Yvon, F. (2015). LIMSI@WMT'15 : Translation Task. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 145–151.

[97] Mariño, J. B., Banchs, R. E., Crego, J. M., de Gispert, A., Lambert, P., Fonollosa, J. A. R., and Costa-jussà, M. R. (2006). N-Gram-based Machine Translation. *Computational Linguistics*, Vol. 32(4), pages 527–549.

[98] Martin, S. C., Liermann, J., and Ney, H. (1997). Adaptive Topic-Dependent Language Modelling Using Word-based Varigrams. In *Proceedings of Eurospeech'97*, pages 1447–1450.

[99] Martzoukos, S. and Monz, C. (2012). Power-Law Distributions for Paraphrases Extracted from Bilingual Corpora. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2–11.

[100] Martzoukos, S., Costa Florêncio, C., and Monz, C. (2013). Investigating Connectivity and Consistency Criteria for Phrase Pair Extraction in Statistical Machine Translation. In *Proceedings of the 13th Meeting on the Mathematics of Language (MoL 13)*, pages 93–101.

[101] Martzoukos, S., Costa Florêncio, C., and Monz, C. (2014). Maximizing Component Quality in Bilingual Word-Aligned Segmentations. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 30–38.

[102] Metzler, D., Hovy, E., and Zhang, C. (2011). An Empirical Evaluation of Data-Driven Paraphrase Generation Techniques. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers*, Vol. 2, pages 546–551.

[103] Mishra, N., Saha Roy, R., Ganguly, N., Laxman, S., and Choudhury, M. (2011). Unsupervised Query Segmentation Using Only Query Logs. In *Proceedings of the 20th International Conference on World Wide Web*, pages 91–92.

[104] Moore, R. C. (2002). Fast and Accurate Sentence Alignment of Bilingual Corpora. In *Proceedings of the 5th Conference of the Association for Machine Translation in the Americas on Machine Translation: From Research to Real Users*, pages 135–144.

[105] Monjardet, B. (1981). Metrics on Partially Ordered Sets – A Survey. *Discrete Mathematics*, Vol. 35(1), pages 173–184.

[106] Neubig, G., Watanabe, T., Sumita, S., Mori, S., and Kawahara, T. (2012). Joint Phrase Alignment and Extraction for Statistical Machine Translation. *Journal of Information Processing*, Vol. 20(2), pages 512–523.

[107] Ney, H., Essen, U., and Kneser, R. (1994). On Structuring Probabilistic Dependences in Stochastic Language Modeling. *Computer, Speech, and Language*, Vol. 8(1), pages 1–38.

[108] Niehues, J., Herrmann, T., Vogel, S., and Waibel, A. (2011). Wider Context by Using Bilingual Language Models in Machine Translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 198–206.

[109] Och, F. J., Tillmann, C., and Ney, H. (1999). Improved Alignment Models for Statistical Machine Translation. In *Proceedings of the Joint Conference of Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP-VLC)*, pages 20–28.

[110] Och, F. J. (2003). Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, Vol. 1, pages 160–167.

[111] Och, F. J. and Ney, H. (2003). A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, Vol. 29(1), pages 19–51.

[112] Onishi, T., Utiyama, M., and Sumita, E. (2011). Paraphrase Lattice for Statistical Machine Translation. *IEICE Transactions on Information and Systems*, Vol. 94(6), pages 1299–1305.

[113] Pagin, P. (2003). Communication and Strong Compositionality. *Journal of Philosophical Logic*, Vol. 32(3), pages 287–322.

[114] Pal, S., Lohar, P., and Naskar, S. K. (2014). Role of Paraphrases in PB-SMT. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 242–253.

[115] Papineni, K., Roukos, S., Ward, T., and Zhu, W. J. (2002). BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318.

[116] Michael, P., Finch, A., and Sumita, E. (2011). Integration of Multiple Bilingually-Trained Segmentation Schemes into Statistical Machine Translation. *IEICE Transactions on Information and Systems*, Vol. 94(3), pages 690–697.

[117] Pelletier, J. (1994). The Principle of Semantic Compositionality. *Topoi*, Vol. 13(1), pages 11–24.

[118] Perman, M., Pitman, J., and Yor, M. (1992). Size-biased Sampling of Poisson Point Processes and Excursions. *Probability Theory and Related Fields*, Vol. 92(1), pages 21–39.

[119] Pitman, J. (2002). Combinatorial Stochastic Processes. *Ecole d'Eté de Probabilités de Saint-Flour XXXII*, number 1875 in Lecture notes in mathematics, Springer.

[120] Pollicott, M. (1993). *Lectures on Ergodic Theory and Pesin Theory on Compact Manifolds*. Vol. 180, Cambridge University Press.

[121] Quirk, C. and Menezes, A. (2006). Do we Need Phrases? Challenging the Conventional Wisdom in Statistical Machine Translation. In *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 9–16.

[122] Rabiner, L. R. (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In *Proceedings of the IEEE*, Vol. 77(2), pages 257-286.

[123] Rao, D., Yarowsky, D., and Callison-Burch, C. (2008). Affinity Measures Based on the Graph Laplacian. In *Proceedings of the 3rd Textgraphs Workshop on Graph-based Algorithms for Natural Language Processing*, pages 41–48.

[124] Ries, K., Buo, F. D., and Waibel, A. (1996). Class Phrase Models for Language Modeling. In *Proceedings of the Fourth International Conference on Spoken Language Processing*, Vol. 1, pages 398–401.

[125] Risvik, K. M., Mikolajewski, T., and Boros, P. (2003). Query Segmentation for Web Search. In *Poster Session in The Twelfth International World Wide Web Conference*.

[126] Robertson, S. (2004). Understanding Inverse Document Frequency: On Theoretical Arguments for IDF. *Journal of Documentation*, Vol. 60(5), pages 503–520.

[127] Rosenfeld, R. (1994). *Adaptive Statistical Language Modeling: A Maximum Entropy Approach.* Ph.D. thesis, Carnegie Mellon University.

[128] Rubinstein, R. Y. (1997). Optimization of Computer Simulation Models with Rare Events. *European Journal of Operations Research*, Vol. 99(1), pages 89–112.

[129] Rubinstein, R. Y. and Kroese, D. P. (2004). *The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation and Machine Learning*. Springer-Verlag, New York.

[130] Sanchis-Trilles, G., Ortiz-Martínez, D., González-Rubio, J., González, J., and Casacuberta, F. (2011). Bilingual Segmentation for Phrasetable Pruning in Statistical Machine Translation. In *Proceedings of the 15th Conference of the European Association for Machine Translation*, pages 257–264.

[131] Sennrich, R. (2012). Perplexity Minimization for Translation Model Domain Adaptation in Statistical Machine Translation. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 539–549.

[132] Servan, C. and Petitrenaud, S. (2012). Calculation of Phrase Probabilities for Statistical Machine Translation by Using Belief Functions. In *Proceedings of COLING 2012: Posters*, pages 1101–1110.

[133] Seymore, K. and Rosenfeld, R. (1996). Scalable Backoff Language Models. In *Proceedings of the Fourth International Conference on Spoken Language Processing*, Vol. 1, pages 232–235.

[134] Shannon, C. E. (1948). A Mathematical Theory of Communication. Bell System Technical Journal, Vol. 27, pages 379–423, 623–656.

[135] Sim, K., Gopalkrishnan, V., Zimek, A., and Cong, G. (2013). A Survey on Enhanced Subspace Clustering. *Data Mining and Knowledge Discovery*, Vol. 26(2), pages 332–397.

[136] Simard, M., Cancedda, N., Cavestro, B., Dymetman, M., Gaussier, E., Goutte, C., and Mauser, A. (2005). Translating With Non-Contiguous Phrases. In

Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, pages 755–762.

[137] Simovici, D. A. and Jaroszewicz, S. (2002). An Axiomatization of Partition Entropy. *IEEE Transactions on Information Theory*, Vol. 48(7), pages 2138–2142.

[138] Simovici, D. A. (2010). Several Remarks on Metrics on Partition Lattices and Their Applications in Data Mining. *Libertas Mathematica*, Vol. 30, pages 19–32.

[139] Siu, M. and Ostendorf, M. (2000). Variable n-Grams and Extensions for Conversational Speech Language Modeling. *IEEE Transactions on Speech and Audio Processing*, Vol. 8(1), pages 63–75.

[140] Sokal, A. D. (2005). The Multivariate Tutte Polynomial (alias Potts Model) for Graphs and Matroids. *Surveys in Combinatorics, London Mathematical Society Lecture Note Series*, Vol. 327, Cambridge University Press, pages 173–226.

[141] Sparck Jones, K. (1972). A Statistical Interpretation of Term Specificity and its Application in Retrieval. *Journal of Documentation*, Vol. 28(1), pages 11–21.

[142] Stanley, R. P. (1997). *Enumerative Combinatorics, Volume 1.* Cambridge University Press, Cambridge, UK.

[143] Stolcke, A. (2000). Entropy-based Pruning of Backoff Language Models. In *Proceedings DARPA Broadcast News Transcription and Understanding Workshop*, pages 270–274.

[144] Stolcke, A. (2002). SRILM – An Extensible Language Modeling Toolkit. In *Proceedings of the International Conference on Spoken Language Processing*, Vol. 2, pages 901–904.

[145] Tan, B. and Peng, F. (2008). Unsupervised Query Segmentation Using Generative Language Models and Wikipedia. In *Proceedings of the 17th International Conference on World Wide Web*, pages 347–356.

[146] Tarjan, R. E. (1974). A Note on Finding the Bridges of a Graph. *Information Processing Letters*, Vol. 2(6), pages 160–161.

[147] Tarski, A. (1956). The Concept of Truth in Formalized Languages. *Logic, Semantics, Metamathematics*, Vol. 2, pages 152–278.

[148] Teh, Y. W. (2006). A Bayesian Interpretation of Interpolated Kneser-Ney. *Technical Report TRA2/06, School of Computing, National University of Singapore*.

[149] Tiedemann, J. (2003). Combining Clues for Word Alignment. In *Proceedings of the 10th Conference on European Chapter of the Association for Computational Linguistics*, Vol. 1, pages 339–346.

[150] Tiedemann, J. (2011). Bitext Alignment. *Synthesis Lectures on Human Language Technologies*, Vol. 4(2), pages 1–165.

[151] Tillmann, C. and Ney, H. (2003). Word Reordering and a Dynamic Programming Beam-Search Algorithm for Statistical Machine Translation. *Computational Linguistics*, Vol. 29(1), pages 97–133.

[152] Tillmann, C. (2008). A Rule-Driven Dynamic Programming Decoder for Statistical MT. In *Proceedings of the Second ACL Workshop on Syntax and Structure in Statistical Translation (SSST-2)*, pages 37–45.

[153] Tu, Z., Liu, Q., and Lin, S. (2013). A Novel Graph-based Compact Representation of Word Alignment. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics - Short Papers*, pages 358–363.

[154] Valiant, L. G. (1979). The Complexity of Enumeration and Reliability Problems. *SIAM Journal on Computing*, Vol. 8(3), pages 410–421.

[155] Welsh, D. J. A. (1997). Approximate Counting. *Surveys in Combinatorics, London Mathematical Society Lecture Notes Series*, Vol. 241, Cambridge University Press, pages 287–324.

[156] Wu, D. (1997). Stochastic Inversion Transduction Grammars and Bilingual Parsing of Parallel Corpora. *Computational Linguistics*, Vol. 23(3), pages 377–403.

[157] Wuebker, J., Mauser, A., and Ney, H. (2010). Training Phrase Translation Models With Leaving-One-Out. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 475–484.

[158] Xiong, D., Zhang M., and Li, H. (2011). A Maximum-Entropy Segmentation Model for Statistical Machine Translation. *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 19(8), pages 2494–2505.

[159] Yamada, K. and Knight, K. (2001). A Syntax-based Statistical Translation Model. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 523–530.

[160] Yule, G. U. (1925). A Mathematical Theory of Evolution, Based on the Conclusions of Dr. J. C. Willis, F.R.S. *Philosophical Transactions of the Royal Society of London*, Series B, Containing Papers of a Biological Character, pages 21–87.

[161] Zadrozny, W. (1994). From Compositional to Systematic Semantics. *Linguistics and Philosophy*, Vol. 17(4), pages 329–342.

[162] Zaslavskiy, M., Dymetman, M., and Cancedda, N. (2009). Phrase-based Statistical Machine Translation as a Traveling Salesman Problem. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 333–341.

[163] Zens, R., Stanton, D., and Xu, P. (2012). A Systematic Comparison of Phrase Table Pruning Techniques. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 972–983.

[164] Zhang, H., Toutanova, K., Quirk, C., and Gao, J. (2013). Beyond Left-to-Right: Multiple Decomposition Structures for SMT. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (HLT-NAACL)*, pages 12–21.

[165] Zhang, Y., Vogel, S., and Waibel, A. (2003). Integrated Phrase Segmentation and Alignment Algorithm for Statistical Machine Translation. In *Proceedings of International Conference on Natural Language Processing and Knowledge Engineering (NLP-KE)*, pages 567–573.

[166] Zhao, S., Niu, C., Zhou, M., Liu, T., and Li, S. (2008). Combining Multiple Resources to Improve SMT-based Paraphrasing Model. In *Proceedings of the 46th*

Annual Meeting of the Association for Computational Linguistics on Human Language Technologies, pages 1021–1029.

[167] Zhao, S., Wang, H., Liu, T., and Li, S. (2008). Pivot Approach for Extracting Paraphrase Patterns from Bilingual Corpora. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies*, pages 780–788.

[168] Zhu, Z., Bernhard, D., and Gurevych, I. (2010). A Monolingual Tree-based Translation Model for Sentence Simplification. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1353–1361.

Samenvatting

Het onderwerp van dit proefschrift zijn de bouwstenen van Statistich Automatisch Vertalen (Statistical Machine Translation, SMT). Er wordt aangetoond dat deze bouwstenen, zinsdelen die verkregen zijn uit tweetalige aligned corpora, een rijkere structuur hebben dan algemeen verondersteld. Een grondige verklaring van het extractiemechanisme toont aan dat de verzameling bouwstenen die het oplevert zich leent voor wiskundige analyse, wat de mogelijkheid biedt tot het ontwikkelen van nieuwe SMT tools en benaderingen. Met dit doel zijn verbanden tussen graaftheorie en waarschijnlijkheidsleer onderzocht om kansfuncties af te leiden voor het opdelen van zinnen in zinsdelen, en voor vertaal-regels. Wat deze regels betreft ondersteunen experimentele resultaten het idee van een statistisch principe van compositionaliteit van vertalingen, wat in de toekomst het onderszoek naar het genereren van data kan bevorderen. Bovendien, aangezien de bestanddelen van compositionaliteit de oorspronkelijke bouwstenen van vertaling (verkregen dmv het trainingsproces) vormen, onderzoeken we of ze eentalige bouwstenen (frasen) generaliseren, en zo ja, welke. Dit leidt tot de identificatie van de rol van puntsgewijs wederzijdse informatie (pointwise mutual information, PMI) als de afstands-metriek over segmentatie-verfijningen. Experimenten tonen aan dat deze gedeeltelijk geordende benadering meer geschikt is dan een standaard taalmodel-benadering voor het vinden van de 'natuurlijke' bouwstenen van eentalige corpora.

Abstract

The subject of investigation of this thesis is the building blocks of translation in Statistical Machine Translation (SMT). We find that these building blocks, namely phrase-level dictionary entries, which are extracted from bilingual aligned corpora (training data), admit richer structure than previously known. A rigorous explanation of the extraction mechanism shows that the resulting set of building blocks is amenable to mathematical investigation with the potential of developing tools and new frameworks for translation. To this end we bridge previously unseen gaps between graph theory and probability theory within SMT in order to derive probability mass functions for phrase-level sentence segmentations and rules of translation. For the latter, experimental results support the claim of a statistical (principle of) compositionality of translation rules which fosters future work on data generation. In addition, since the constituents of composition are the original building blocks of translation, as extracted from the training process, we investigate whether they generalize monolingual building blocks (phrases), and if so, of what type. This leads to identifying the role of pointwise mutual information as the distance metric on segmentation refinements. Experiments show that such a partially ordered framework is more appropriate than a standard language model approach for finding the 'natural' building blocks of monolingual corpora.

SIKS Dissertation Series

1998

1 Johan van den Akker (CWI) *DEGAS - An Active, Temporal Database of Autonomous Objects*
2 Floris Wiesman (UM) *Information Retrieval by Graphically Browsing Meta-Information*
3 Ans Steuten (TUD) *A Contribution to the Linguistic Analysis of Business Conversations within the Language/Action Perspective*
4 Dennis Breuker (UM) *Memory versus Search in Games*
5 Eduard W. Oskamp (RUL) *Computerondersteuning bij Straftoemeting*

1999

1 Mark Sloof (VU) *Physiology of Quality Change Modelling; Automated modelling of Quality Change of Agricultural Products*
2 Rob Potharst (EUR) *Classification using decision trees and neural nets*
3 Don Beal (UM) *The Nature of Minimax Search*
4 Jacques Penders (UM) *The practical Art of Moving Physical Objects*
5 Aldo de Moor (KUB) *Empowering Communities: A Method for the Legitimate User-Driven Specification of Network Information Systems*
6 Niek J.E. Wijngaards (VU) *Re-design of compositional systems*
7 David Spelt (UT) *Verification support for object database design*
8 Jacques H.J. Lenting (UM) *Informed Gambling: Conception and Analysis of a Multi-Agent Mechanism for Discrete Reallocation.*

2000

1 Frank Niessink (VU) *Perspectives on Improving Software Maintenance*
2 Koen Holtman (TUE) *Prototyping of CMS Storage Management*
3 Carolien M.T. Metselaar (UVA) *Sociaal-organisatorische gevolgen van kennistechnologie; een procesbenadering en actorperspectief.*
4 Geert de Haan (VU) *ETAG, A Formal Model of Competence Knowledge for User Interface Design*
5 Ruud van der Pol (UM) *Knowledge-based Query Formulation in Information Retrieval.*
6 Rogier van Eijk (UU) *Programming Languages for Agent Communication*
7 Niels Peek (UU) *Decision-theoretic Planning of Clinical Patient Management*
8 Veerle Coupé (EUR) *Sensitivity Analyis of Decision-Theoretic Networks*
9 Florian Waas (CWI) *Principles of Probabilistic Query Optimization*
10 Niels Nes (CWI) *Image Database Management System Design Considerations, Algorithms and Architecture*
11 Jonas Karlsson (CWI) *Scalable Distributed Data Structures for Database Management*

2001

1 Silja Renooij (UU) *Qualitative Approaches to Quantifying Probabilistic Networks*
2 Koen Hindriks (UU) *Agent Programming Languages: Programming with Mental Models*
3 Maarten van Someren (UvA) *Learning as problem solving*

5 Jacco van Ossenbruggen (VU) *Processing Structured Hypermedia: A Matter of Style*
6 Martijn van Welie (VU) *Task-based User Interface Design*
7 Bastiaan Schonhage (VU) *Diva: Architectural Perspectives on Information Visualization*
9 Pieter Jan 't Hoen (RUL) *Towards Distributed Development of Large Object-Oriented Models, Views of Packages as Classes*
10 Maarten Sierhuis (UvA) *Modeling and Simulating Work Practice BRAHMS: a multiagent modeling and simulation language for work practice analysis and design*
11 Tom M. van Engers (VU) *Knowledge Management: The Role of Mental Models in Business Systems Design*

2002

1 Nico Lassing (VU) *Architecture-Level Modifiability Analysis*
2 Roelof van Zwol (UT) *Modelling and searching web-based document collections*
4 Juan Roberto Castelo Valdueza (UU) *The Discrete Acyclic Digraph Markov Model in Data Mining*
6 Laurens Mommers (UL) *Applied legal epistemology; Building a knowledge-based ontology of the legal domain*
9 Willem-Jan van den Heuvel (KUB) *Integrating Modern Business Applications with Objectified Legacy Systems*
12 Albrecht Schmidt (UVA) *Processing XML in Database Systems*
13 Hongjing Wu (TUE) *A Reference Architecture for Adaptive Hypermedia Applications*
14 Wieke de Vries (UU) *Agent Interaction: Abstract Approaches to Modelling, Programming and Verifying Multi-Agent Systems*
15 Rik Eshuis (UT) *Semantics and Verification of UML Activity Diagrams for Workflow Modelling*
16 Pieter van Langen (VU) *The Anatomy of Design: Foundations, Models and Applications*
17 Stefan Manegold (UVA) *Understanding, Modeling, and Improving Main-Memory Database Performance*

2003

1 Heiner Stuckenschmidt (VU) *Ontology-Based Information Sharing in Weakly Structured Environments*
3 Martijn Schuemie (TUD) *Human-Computer Interaction and Presence in Virtual Reality Exposure Therapy*
4 Milan Petkovic (UT) *Content-Based Video Retrieval Supported by Database Technology*
5 Jos Lehmann (UVA) *Causation in Artificial Intelligence and Law - A modelling approach*
6 Boris van Schooten (UT) *Development and specification of virtual environments*
7 Machiel Jansen (UvA) *Formal Explorations of Knowledge Intensive Tasks*
8 Yongping Ran (UM) *Repair Based Scheduling*
9 Rens Kortmann (UM) *The resolution of visually guided behaviour*
11 Simon Keizer (UT) *Reasoning under Uncertainty in Natural Language Dialogue using Bayesian Networks*
12 Roeland Ordelman (UT) *Dutch speech recognition in multimedia information retrieval*
13 Jeroen Donkers (UM) *Nosce Hostem - Searching with Opponent Models*
14 Stijn Hoppenbrouwers (KUN) *Freezing Language: Conceptualisation Processes across ICT-Supported Organisations*
15 Mathijs de Weerdt (TUD) *Plan Merging in Multi-Agent Systems*
16 Menzo Windhouwer (CWI) *Feature Grammar Systems - Incremental Maintenance of Indexes to Digital Media Warehouses*
17 David Jansen (UT) *Extensions of Statecharts with Probability, Time, and Stochastic Timing*
18 Levente Kocsis (UM) *Learning Search Decisions*

2004

1 Virginia Dignum (UU) *A Model for Organizational Interaction: Based on Agents, Founded in Logic*
2 Lai Xu (UvT) *Monitoring Multi-party Contracts for E-business*
3 Perry Groot (VU) *A Theoretical and Empirical Analysis of Approximation in Symbolic Problem Solving*
4 Chris van Aart (UVA) *Organizational Principles for Multi-Agent Architectures*
5 Viara Popova (EUR) *Knowledge discovery and monotonicity*
6 Bart-Jan Hommes (TUD) *The Evaluation of Business Process Modeling Techniques*
7 Elise Boltjes (UM) *Voorbeeldig onderwijs; voorbeeldgestuurd onderwijs, een opstap naar abstract denken, vooral voor meisjes*
8 Joop Verbeek (UM) *Politie en de Nieuwe Internationale Informatiemarkt, Grensregionale politiï£¡le gegevensuitwisseling en digitale expertise*
10 Suzanne Kabel (UVA) *Knowledge-rich indexing of learning-objects*
11 Michel Klein (VU) *Change Management for Distributed Ontologies*
13 Wojciech Jamroga (UT) *Using Multiple Models of Reality: On Agents who Know how to Play*
14 Paul Harrenstein (UU) *Logic in Conflict. Logical Explorations in Strategic Equilibrium*
15 Arno Knobbe (UU) *Multi-Relational Data Mining*
16 Federico Divina (VU) *Hybrid Genetic Relational Search for Inductive Learning*
17 Mark Winands (UM) *Informed Search in Complex Games*
18 Vania Bessa Machado (UvA) *Supporting the Construction of Qualitative Knowledge Models*
19 Thijs Westerveld (UT) *Using generative probabilistic models for multimedia retrieval*
20 Madelon Evers (Nyenrode) *Learning from Design: facilitating multidisciplinary design teams*

2005

1 Floor Verdenius (UVA) *Methodological Aspects of Designing Induction-Based Applications*
2 Erik van der Werf (UM) *AI techniques for the game of Go*
3 Franc Grootjen (RUN) *A Pragmatic Approach to the Conceptualisation of Language*
4 Nirvana Meratnia (UT) *Towards Database Support for Moving Object data*
5 Gabriel Infante-Lopez (UVA) *Two-Level Probabilistic Grammars for Natural Language Parsing*
6 Pieter Spronck (UM) *Adaptive Game AI*
7 Flavius Frasincar (TUE) *Hypermedia Presentation Generation for Semantic Web Information Systems*
8 Richard Vdovjak (TUE) *A Model-driven Approach for Building Distributed Ontology-based Web Applications*
9 Jeen Broekstra (VU) *Storage, Querying and Inferencing for Semantic Web Languages*
10 Anders Bouwer (UVA) *Explaining Behaviour: Using Qualitative Simulation in Interactive Learning Environments*
11 Elth Ogston (VU) *Agent Based Matchmaking and Clustering - A Decentralized Approach to Search*
12 Csaba Boer (EUR) *Distributed Simulation in Industry*
13 Fred Hamburg (UL) *Een Computermodel voor het Ondersteunen van Euthanasiebeslissingen*
14 Borys Omelayenko (VU) *Web-Service configuration on the Semantic Web; Exploring how semantics meets pragmatics*
15 Tibor Bosse (VU) *Analysis of the Dynamics of Cognitive Processes*
16 Joris Graaumans (UU) *Usability of XML Query Languages*
17 Boris Shishkov (TUD) *Software Specification Based on Re-usable Business Components*
18 Danielle Sent (UU) *Test-selection strategies for probabilistic networks*
19 Michel van Dartel (UM) *Situated Representation*
20 Cristina Coteanu (UL) *Cyber Consumer Law, State of the Art and Perspectives*
21 Wijnand Derks (UT) *Improving Concurrency and Recovery in Database Systems by Exploiting Application Semantics*

2006

1 Samuil Angelov (TUE) *Foundations of B2B Electronic Contracting*
2 Cristina Chisalita (VU) *Contextual issues in the design and use of information technology in organizations*
3 Noor Christoph (UVA) *The role of metacognitive skills in learning to solve problems*
4 Marta Sabou (VU) *Building Web Service Ontologies*
5 Cees Pierik (UU) *Validation Techniques for Object-Oriented Proof Outlines*
6 Ziv Baida (VU) *Software-aided Service Bundling - Intelligent Methods & Tools for Graphical Service Modeling*
7 Marko Smiljanic (UT) *XML schema matching – balancing efficiency and effectiveness by means of clustering*
8 Eelco Herder (UT) *Forward, Back and Home Again - Analyzing User Behavior on the Web*
9 Mohamed Wahdan (UM) *Automatic Formulation of the Auditor's Opinion*
10 Ronny Siebes (VU) *Semantic Routing in Peer-to-Peer Systems*
11 Joeri van Ruth (UT) *Flattening Queries over Nested Data Types*
12 Bert Bongers (VU) *Interactivation - Towards an e-cology of people, our technological environment, and the arts*
13 Henk-Jan Lebbink (UU) *Dialogue and Decision Games for Information Exchanging Agents*
14 Johan Hoorn (VU) *Software Requirements: Update, Upgrade, Redesign - towards a Theory of Requirements Change*
15 Rainer Malik (UU) *CONAN: Text Mining in the Biomedical Domain*
16 Carsten Riggelsen (UU) *Approximation Methods for Efficient Learning of Bayesian Networks*
17 Stacey Nagata (UU) *User Assistance for Multitasking with Interruptions on a Mobile Device*
18 Valentin Zhizhkun (UVA) *Graph transformation for Natural Language Processing*
19 Birna van Riemsdijk (UU) *Cognitive Agent Programming: A Semantic Approach*
20 Marina Velikova (UvT) *Monotone models for prediction in data mining*
21 Bas van Gils (RUN) *Aptness on the Web*
22 Paul de Vrieze (RUN) *Fundaments of Adaptive Personalisation*
23 Ion Juvina (UU) *Development of Cognitive Model for Navigating on the Web*
24 Laura Hollink (VU) *Semantic Annotation for Retrieval of Visual Resources*
25 Madalina Drugan (UU) *Conditional log-likelihood MDL and Evolutionary MCMC*
26 Vojkan Mihajlovic (UT) *Score Region Algebra: A Flexible Framework for Structured Information Retrieval*
27 Stefano Bocconi (CWI) *Vox Populi: generating video documentaries from semantically annotated media repositories*
28 Borkur Sigurbjornsson (UVA) *Focused Information Access using XML Element Retrieval*

2007

1 Kees Leune (UvT) *Access Control and Service-Oriented Architectures*
2 Wouter Teepe (RUG) *Reconciling Information Exchange and Confidentiality: A Formal Approach*
3 Peter Mika (VU) *Social Networks and the Semantic Web*
4 Jurriaan van Diggelen (UU) *Achieving Semantic Interoperability in Multi-agent Systems: a dialogue-based approach*
5 Bart Schermer (UL) *Software Agents, Surveillance, and the Right to Privacy: a Legislative Framework for Agent-enabled Surveillance*
6 Gilad Mishne (UVA) *Applied Text Analytics for Blogs*
7 Natasa Jovanovic' (UT) *To Whom It May Concern - Addressee Identification in Face-to-Face Meetings*
8 Mark Hoogendoorn (VU) *Modeling of Change in Multi-Agent Organizations*
9 David Mobach (VU) *Agent-Based Mediated Service Negotiation*
10 Huib Aldewereld (UU) *Autonomy vs. Conformity: an Institutional Perspective on Norms and Protocols*
11 Natalia Stash (TUE) *Incorporating Cognitive/Learning Styles in a General-Purpose Adaptive Hypermedia System*

12 Marcel van Gerven (RUN) *Bayesian Networks for Clinical Decision Support: A Rational Approach to Dynamic Decision-Making under Uncertainty*
13 Rutger Rienks (UT) *Meetings in Smart Environments; Implications of Progressing Technology*
14 Niek Bergboer (UM) *Context-Based Image Analysis*
15 Joyca Lacroix (UM) *NIM: a Situated Computational Memory Model*
16 Davide Grossi (UU) *Designing Invisible Handcuffs. Formal investigations in Institutions and Organizations for Multi-agent Systems*
17 Theodore Charitos (UU) *Reasoning with Dynamic Networks in Practice*
19 David Levy (UM) *Intimate relationships with artificial partners*
20 Slinger Jansen (UU) *Customer Configuration Updating in a Software Supply Network*
21 Karianne Vermaas (UU) *Fast diffusion and broadening use: A research on residential adoption and usage of broadband internet in the Netherlands between 2001 and 2005*
22 Zlatko Zlatev (UT) *Goal-oriented design of value and process models from patterns*
23 Peter Barna (TUE) *Specification of Application Logic in Web Information Systems*
24 Georgina Ramírez Camps (CWI) *Structural Features in XML Retrieval*
25 Joost Schalken (VU) *Empirical Investigations in Software Process Improvement*

2008

1 Katalin Boer-Sorbán (EUR) *Agent-Based Simulation of Financial Markets: A modular, continuous-time approach*
2 Alexei Sharpanskykh (VU) *On Computer-Aided Methods for Modeling and Analysis of Organizations*
3 Vera Hollink (UVA) *Optimizing hierarchical menus: a usage-based approach*
4 Ander de Keijzer (UT) *Management of Uncertain Data - towards unattended integration*
5 Bela Mutschler (UT) *Modeling and simulating causal dependencies on process-aware information systems from a cost perspective*
6 Arjen Hommersom (RUN) *On the Application of Formal Methods to Clinical Guidelines, an Artificial Intelligence Perspective*
7 Peter van Rosmalen (OU) *Supporting the tutor in the design and support of adaptive e-learning*
8 Janneke Bolt (UU) *Bayesian Networks: Aspects of Approximate Inference*
9 Christof van Nimwegen (UU) *The paradox of the guided user: assistance can be counter-effective*
10 Wauter Bosma (UT) *Discourse oriented summarization*
11 Vera Kartseva (VU) *Designing Controls for Network Organizations: A Value-Based Approach*
12 Jozsef Farkas (RUN) *A Semiotically Oriented Cognitive Model of Knowledge Representation*
13 Caterina Carraciolo (UVA) *Topic Driven Access to Scientific Handbooks*
14 Arthur van Bunningen (UT) *Context-Aware Querying; Better Answers with Less Effort*
15 Martijn van Otterlo (UT) *The Logic of Adaptive Behavior: Knowledge Representation and Algorithms for the Markov Decision Process Framework in First-Order Domains.*
16 Henriette van Vugt (VU) *Embodied agents from a user's perspective*
17 Martin Op 't Land (TUD) *Applying Architecture and Ontology to the Splitting and Allying of Enterprises*
18 Guido de Croon (UM) *Adaptive Active Vision*
19 Henning Rode (UT) *From Document to Entity Retrieval: Improving Precision and Performance of Focused Text Search*
20 Rex Arendsen (UVA) *Geen bericht, goed bericht. Een onderzoek naar de effecten van de introductie van elektronisch berichtenverkeer met de overheid op de administratieve lasten van bedrijven.*
21 Krisztian Balog (UVA) *People Search in the Enterprise*
22 Henk Koning (UU) *Communication of IT-Architecture*
23 Stefan Visscher (UU) *Bayesian network models for the management of ventilator-associated pneumonia*
24 Zharko Aleksovski (VU) *Using background knowledge in ontology matching*
25 Geert Jonker (UU) *Efficient and Equitable Exchange in Air Traffic Management Plan Repair using Spender-signed Currency*
26 Marijn Huijbregts (UT) *Segmentation, Diarization and Speech Transcription: Surprise Data Unraveled*
27 Hubert Vogten (OU) *Design and Implementation Strategies for IMS Learning Design*
28 Ildiko Flesch (RUN) *On the Use of Independence Relations in Bayesian Networks*
29 Dennis Reidsma (UT) *Annotations and Subjective Machines - Of Annotators, Embodied Agents, Users, and Other Humans*
30 Wouter van Atteveldt (VU) *Semantic Network Analysis: Techniques for Extracting, Representing and Querying Media Content*
31 Loes Braun (UM) *Pro-Active Medical Information Retrieval*
32 Trung H. Bui (UT) *Toward Affective Dialogue Management using Partially Observable Markov Decision Processes*
33 Frank Terpstra (UVA) *Scientific Workflow Design; theoretical and practical issues*
34 Jeroen de Knijf (UU) *Studies in Frequent Tree Mining*
35 Ben Torben Nielsen (UvT) *Dendritic morphologies: function shapes structure*

2009

1 Rasa Jurgelenaite (RUN) *Symmetric Causal Independence Models*
2 Willem Robert van Hage (VU) *Evaluating Ontology-Alignment Techniques*
3 Hans Stol (UvT) *A Framework for Evidence-based Policy Making Using IT*
4 Josephine Nabukenya (RUN) *Improving the Quality of Organisational Policy Making using Collaboration Engineering*
5 Sietse Overbeek (RUN) *Bridging Supply and Demand for Knowledge Intensive Tasks - Based on Knowledge, Cognition, and Quality*
6 Muhammad Subianto (UU) *Understanding Classification*
7 Ronald Poppe (UT) *Discriminative Vision-Based Recovery and Recognition of Human Motion*

8 Volker Nannen (VU) *Evolutionary Agent-Based Policy Analysis in Dynamic Environments*
9 Benjamin Kanagwa (RUN) *Design, Discovery and Construction of Service-oriented Systems*
10 Jan Wielemaker (UVA) *Logic programming for knowledge-intensive interactive applications*
11 Alexander Boer (UVA) *Legal Theory, Sources of Law & the Semantic Web*
12 Peter Massuthe (TUE, Humboldt-Universitaet zu Berlin) *Operating Guidelines for Services*
13 Steven de Jong (UM) *Fairness in Multi-Agent Systems*
14 Maksym Korotkiy (VU) *From ontology-enabled services to service-enabled ontologies (making ontologies work in e-science with ONTO-SOA)*
15 Rinke Hoekstra (UVA) *Ontology Representation - Design Patterns and Ontologies that Make Sense*
16 Fritz Reul (UvT) *New Architectures in Computer Chess*
17 Laurens van der Maaten (UvT) *Feature Extraction from Visual Data*
18 Fabian Groffen (CWI) *Armada, An Evolving Database System*
19 Valentin Robu (CWI) *Modeling Preferences, Strategic Reasoning and Collaboration in Agent-Mediated Electronic Markets*
20 Bob van der Vecht (UU) *Adjustable Autonomy: Controling Influences on Decision Making*
21 Stijn Vanderlooy (UM) *Ranking and Reliable Classification*
22 Pavel Serdyukov (UT) *Search For Expertise: Going beyond direct evidence*
23 Peter Hofgesang (VU) *Modelling Web Usage in a Changing Environment*
24 Annerieke Heuvelink (VUA) *Cognitive Models for Training Simulations*
25 Alex van Ballegooij (CWI) *"RAM: Array Database Management through Relational Mapping"*
26 Fernando Koch (UU) *An Agent-Based Model for the Development of Intelligent Mobile Services*
27 Christian Glahn (OU) *Contextual Support of social Engagement and Reflection on the Web*
28 Sander Evers (UT) *Sensor Data Management with Probabilistic Models*
29 Stanislav Pokraev (UT) *Model-Driven Semantic Integration of Service-Oriented Applications*
30 Marcin Zukowski (CWI) *Balancing vectorized query execution with bandwidth-optimized storage*
31 Sofiya Katrenko (UVA) *A Closer Look at Learning Relations from Text*
32 Rik Farenhorst (VU) and Remco de Boer (VU) *Architectural Knowledge Management: Supporting Architects and Auditors*
33 Khiet Truong (UT) *How Does Real Affect Affect Affect Recognition In Speech?*
34 Inge van de Weerd (UU) *Advancing in Software Product Management: An Incremental Method Engineering Approach*
35 Wouter Koelewijn (UL) *Privacy en Politiegegevens; Over geautomatiseerde normatieve informatie-uitwisseling*
36 Marco Kalz (OUN) *Placement Support for Learners in Learning Networks*
37 Hendrik Drachsler (OUN) *Navigation Support for Learners in Informal Learning Networks*
38 Riina Vuorikari (OU) *Tags and self-organisation: a metadata ecology for learning resources in a multilingual context*
39 Christian Stahl (TUE, Humboldt-Universitaet zu Berlin) *Service Substitution – A Behavioral Approach Based on Petri Nets*
40 Stephan Raaijmakers (UvT) *Multinomial Language Learning: Investigations into the Geometry of Language*
41 Igor Berezhnyy (UvT) *Digital Analysis of Paintings*
42 Toine Bogers (UvT) *Recommender Systems for Social Bookmarking*
43 Virginia Nunes Leal Franqueira (UT) *Finding Multi-step Attacks in Computer Networks using Heuristic Search and Mobile Ambients*
44 Roberto Santana Tapia (UT) *Assessing Business-IT Alignment in Networked Organizations*
45 Jilles Vreeken (UU) *Making Pattern Mining Useful*
46 Loredana Afanasiev (UvA) *Querying XML: Benchmarks and Recursion*

2010

1 Matthijs van Leeuwen (UU) *Patterns that Matter*
2 Ingo Wassink (UT) *Work flows in Life Science*
3 Joost Geurts (CWI) *A Document Engineering Model and Processing Framework for Multimedia documents*
4 Olga Kulyk (UT) *Do You Know What I Know? Situational Awareness of Co-located Teams in Multidisplay Environments*
5 Claudia Hauff (UT) *Predicting the Effectiveness of Queries and Retrieval Systems*
6 Sander Bakkes (UvT) *Rapid Adaptation of Video Game AI*
7 Wim Fikkert (UT) *Gesture interaction at a Distance*
8 Krzysztof Siewicz (UL) *Towards an Improved Regulatory Framework of Free Software. Protecting user freedoms in a world of software communities and eGovernments*
9 Hugo Kielman (UL) *A Politiele gegevensverwerking en Privacy, Naar een effectieve waarborging*
10 Rebecca Ong (UL) *Mobile Communication and Protection of Children*
11 Adriaan Ter Mors (TUD) *The world according to MARP: Multi-Agent Route Planning*
12 Susan van den Braak (UU) *Sensemaking software for crime analysis*
13 Gianluigi Folino (RUN) *High Performance Data Mining using Bio-inspired techniques*
14 Sander van Splunter (VU) *Automated Web Service Reconfiguration*
15 Lianne Bodenstaff (UT) *Managing Dependency Relations in Inter-Organizational Models*
16 Sicco Verwer (TUD) *Efficient Identification of Timed Automata, theory and practice*
17 Spyros Kotoulas (VU) *Scalable Discovery of Networked Resources: Algorithms, Infrastructure, Applications*
18 Charlotte Gerritsen (VU) *Caught in the Act: Investigating Crime by Agent-Based Simulation*
19 Henriette Cramer (UvA) *People's Responses to Autonomous and Adaptive Systems*
20 Ivo Swartjes (UT) *Whose Story Is It Anyway? How Improv Informs Agency and Authorship of Emergent Narrative*
21 Harold van Heerde (UT) *Privacy-aware data management by means of data degradation*
22 Michiel Hildebrand (CWI) *End-user Support for Access to Heterogeneous Linked Data*
23 Bas Steunebrink (UU) *The Logical Structure of Emotions*
24 Dmytro Tykhonov *Designing Generic and Efficient Negotiation Strategies*
25 Zulfiqar Ali Memon (VU) *Modelling Human-Awareness for Ambient Agents: A Human Mindreading Perspective*

26 Ying Zhang (CWI) *XRPC: Efficient Distributed Query Processing on Heterogeneous XQuery Engines*

27 Marten Voulon (UL) *Automatisch contracteren*

28 Arne Koopman (UU) *Characteristic Relational Patterns*

29 Stratos Idreos (CWI) *Database Cracking: Towards Auto-tuning Database Kernels*

30 Marieke van Erp (UvT) *Accessing Natural History - Discoveries in data cleaning, structuring, and retrieval*

31 Victor de Boer (UVA) *Ontology Enrichment from Heterogeneous Sources on the Web*

32 Marcel Hiel (UvT) *An Adaptive Service Oriented Architecture: Automatically solving Interoperability Problems*

33 Robin Aly (UT) *Modeling Representation Uncertainty in Concept-Based Multimedia Retrieval*

34 Teduh Dirgahayu (UT) *Interaction Design in Service Compositions*

35 Dolf Trieschnigg (UT) *Proof of Concept: Concept-based Biomedical Information Retrieval*

36 Jose Janssen (OU) *Paving the Way for Lifelong Learning; Facilitating competence development through a learning path specification*

37 Niels Lohmann (TUE) *Correctness of services and their composition*

38 Dirk Fahland (TUE) *From Scenarios to components*

39 Ghazanfar Farooq Siddiqui (VU) *Integrative modeling of emotions in virtual agents*

40 Mark van Assem (VU) *Converting and Integrating Vocabularies for the Semantic Web*

41 Guillaume Chaslot (UM) *Monte-Carlo Tree Search*

42 Sybren de Kinderen (VU) *Needs-driven service bundling in a multi-supplier setting - the computational e3-service approach*

43 Peter van Kranenburg (UU) *A Computational Approach to Content-Based Retrieval of Folk Song Melodies*

44 Pieter Bellekens (TUE) *An Approach towards Context-sensitive and User-adapted Access to Heterogeneous Data Sources, Illustrated in the Television Domain*

45 Vasilios Andrikopoulos (UvT) *A theory and model for the evolution of software services*

46 Vincent Pijpers (VU) *e3alignment: Exploring Inter-Organizational Business-ICT Alignment*

47 Chen Li (UT) *Mining Process Model Variants: Challenges, Techniques, Examples*

48 (Withdrawn)

49 Jahn-Takeshi Saito (UM) *Solving difficult game positions*

50 Bouke Huurnink (UVA) *Search in Audiovisual Broadcast Archives*

51 Alia Khairia Amin (CWI) *Understanding and supporting information seeking tasks in multiple sources*

52 Peter-Paul van Maanen (VU) *Adaptive Support for Human-Computer Teams: Exploring the Use of Cognitive Models of Trust and Attention*

53 Edgar Meij (UVA) *Combining Concepts and Language Models for Information Access*

2011

1 Botond Cseke (RUN) *Variational Algorithms for Bayesian Inference in Latent Gaussian Models*

2 Nick Tinnemeier (UU) *Organizing Agent Organizations. Syntax and Operational Semantics of an Organization-Oriented Programming Language*

3 Jan Martijn van der Werf (TUE) *Compositional Design and Verification of Component-Based Information Systems*

4 Hado van Hasselt (UU) *Insights in Reinforcement Learning; Formal analysis and empirical evaluation of temporal-difference*

5 Base van der Raadt (VU) *Enterprise Architecture Coming of Age - Increasing the Performance of an Emerging Discipline.*

6 Yiwen Wang (TUE) *Semantically-Enhanced Recommendations in Cultural Heritage*

7 Yujia Cao (UT) *Multimodal Information Presentation for High Load Human Computer Interaction*

8 Nieske Vergunst (UU) *BDI-based Generation of Robust Task-Oriented Dialogues*

9 Tim de Jong (OU) *Contextualised Mobile Media for Learning*

10 Bart Bogaert (UvT) *Cloud Content Contention*

11 Dhaval Vyas (UT) *Designing for Awareness: An Experience-focused HCI Perspective*

12 Carmen Bratosin (TUE) *Grid Architecture for Distributed Process Mining*

13 Xiaoyu Mao (UvT) *Airport under Control. Multiagent Scheduling for Airport Ground Handling*

14 Milan Lovric (EUR) *Behavioral Finance and Agent-Based Artificial Markets*

15 Marijn Koolen (UvA) *The Meaning of Structure: the Value of Link Evidence for Information Retrieval*

16 Maarten Schadd (UM) *Selective Search in Games of Different Complexity*

17 Jiyin He (UVA) *Exploring Topic Structure: Coherence, Diversity and Relatedness*

18 Mark Ponsen (UM) *Strategic Decision-Making in complex games*

19 Ellen Rusman (OU) *The Mind ’ s Eye on Personal Profiles*

20 Qing Gu (VU) *Guiding service-oriented software engineering - A view-based approach*

21 Linda Terlouw (TUD) *Modularization and Specification of Service-Oriented Systems*

22 Junte Zhang (UVA) *System Evaluation of Archival Description and Access*

23 Wouter Weerkamp (UVA) *Finding People and their Utterances in Social Media*

24 Herwin van Welbergen (UT) *Behavior Generation for Interpersonal Coordination with Virtual Humans On Specifying, Scheduling and Realizing Multimodal Virtual Human Behavior*

25 Syed Waqar ul Qounain Jaffry (VU) *Analysis and Validation of Models for Trust Dynamics*

26 Matthijs Aart Pontier (VU) *Virtual Agents for Human Communication - Emotion Regulation and Involvement-Distance Trade-Offs in Embodied Conversational Agents and Robots*

27 Aniel Bhulai (VU) *Dynamic website optimization through autonomous management of design patterns*

28 Rianne Kaptein (UVA) *Effective Focused Retrieval by Exploiting Query Context and Document Structure*

29 Faisal Kamiran (TUE) *Discrimination-aware Classification*

30 Egon van den Broek (UT) *Affective Signal Processing (ASP): Unraveling the mystery of emotions*

31 Ludo Waltman (EUR) *Computational and Game-Theoretic Approaches for Modeling Bounded Rationality*

32 Nees-Jan van Eck (EUR) *Methodological Advances in Bibliometric Mapping of Science*

33 Tom van der Weide (UU) *Arguing to Motivate Decisions*

34 Paolo Turrini (UU) *Strategic Reasoning in Interdependence: Logical and Game-theoretical Investigations*
35 Maaike Harbers (UU) *Explaining Agent Behavior in Virtual Training*
36 Erik van der Spek (UU) *Experiments in serious game design: a cognitive approach*
37 Adriana Burlutiu (RUN) *Machine Learning for Pairwise Data, Applications for Preference Learning and Supervised Network Inference*
38 Nyree Lemmens (UM) *Bee-inspired Distributed Optimization*
39 Joost Westra (UU) *Organizing Adaptation using Agents in Serious Games*
40 Viktor Clerc (VU) *Architectural Knowledge Management in Global Software Development*
41 Luan Ibraimi (UT) *Cryptographically Enforced Distributed Data Access Control*
42 Michal Sindlar (UU) *Explaining Behavior through Mental State Attribution*
43 Henk van der Schuur (UU) *Process Improvement through Software Operation Knowledge*
44 Boris Reuderink (UT) *Robust Brain-Computer Interfaces*
45 Herman Stehouwer (UvT) *Statistical Language Models for Alternative Sequence Selection*
46 Beibei Hu (TUD) *Towards Contextualized Information Delivery: A Rule-based Architecture for the Domain of Mobile Police Work*
47 Azizi Bin Ab Aziz (VU) *Exploring Computational Models for Intelligent Support of Persons with Depression*
48 Mark Ter Maat (UT) *Response Selection and Turn-taking for a Sensitive Artificial Listening Agent*
49 Andreea Niculescu (UT) *Conversational interfaces for task-oriented spoken dialogues: design aspects influencing interaction quality*

2012

1 Terry Kakeeto (UvT) *Relationship Marketing for SMEs in Uganda*
2 Muhammad Umair (VU) *Adaptivity, emotion, and Rationality in Human and Ambient Agent Models*
3 Adam Vanya (VU) *Supporting Architecture Evolution by Mining Software Repositories*
4 Jurriaan Souer (UU) *Development of Content Management System-based Web Applications*
6 Wolfgang Reinhardt (OU) *Awareness Support for Knowledge Workers in Research Networks*
7 Rianne van Lambalgen (VU) *When the Going Gets Tough: Exploring Agent-based Models of Human Performance under Demanding Conditions*
8 Gerben de Vries (UVA) *Kernel Methods for Vessel Trajectories*
9 Ricardo Neisse (UT) *Trust and Privacy Management Support for Context-Aware Service Platforms*
10 David Smits (TUE) *Towards a Generic Distributed Adaptive Hypermedia Environment*
11 J.C.B. Rantham Prabhakara (TUE) *Process Mining in the Large: Preprocessing, Discovery, and Diagnostics*
12 Kees van der Sluijs (TUE) *Model Driven Design and Data Integration in Semantic Web Information Systems*
13 Suleman Shahid (UvT) *Fun and Face: Exploring non-verbal expressions of emotion during playful interactions*
14 Evgeny Knutov (TUE) *Generic Adaptation Framework for Unifying Adaptive Web-based Systems*
15 Natalie van der Wal (VU) *Social Agents. Agent-Based Modelling of Integrated Internal and Social Dynamics of Cognitive and Affective Processes.*
16 Fiemke Both (VU) *Helping people by understanding them - Ambient Agents supporting task execution and depression treatment*
17 Amal Elgammal (UvT) *Towards a Comprehensive Framework for Business Process Compliance*
18 Eltjo Poort (VU) *Improving Solution Architecting Practices*
19 Helen Schonenberg (TUE) *What's Next? Operational Support for Business Process Execution*
20 Ali Bahramisharif (RUN) *Covert Visual Spatial Attention, a Robust Paradigm for Brain-Computer Interfacing*
21 Roberto Cornacchia (TUD) *Querying Sparse Matrices for Information Retrieval*
22 Thijs Vis (UvT) *Intelligence, politie en veiligheidsdienst: verenigbare grootheden?*
23 Christian Muehl (UT) *Toward Affective Brain-Computer Interfaces: Exploring the Neurophysiology of Affect during Human Media Interaction*
24 Laurens van der Werff (UT) *Evaluation of Noisy Transcripts for Spoken Document Retrieval*
25 Silja Eckartz (UT) *Managing the Business Case Development in Inter-Organizational IT Projects: A Methodology and its Application*
26 Emile de Maat (UVA) *Making Sense of Legal Text*
27 Hayrettin Gurkok (UT) *Mind the Sheep! User Experience Evaluation & Brain-Computer Interface Games*
28 Nancy Pascall (UvT) *Engendering Technology Empowering Women*
29 Almer Tigelaar (UT) *Peer-to-Peer Information Retrieval*
30 Alina Pommeranz (TUD) *Designing Human-Centered Systems for Reflective Decision Making*
33 Rory Sie (OUN) *Coalitions in Cooperation Networks (COCOON)*
34 Pavol Jancura (RUN) *Evolutionary analysis in PPI networks and applications*
35 Evert Haasdijk (VU) *Never Too Old To Learn – On-line Evolution of Controllers in Swarm- and Modular Robotics*
36 Denis Ssebugwawo (RUN) *Analysis and Evaluation of Collaborative Modeling Processes*
37 Agnes Nakakawa (RUN) *A Collaboration Process for Enterprise Architecture Creation*
38 Selmar Smit (VU) *Parameter Tuning and Scientific Testing in Evolutionary Algorithms*
42 Dominique Verpoorten (OU) *Reflection Amplifiers in self-regulated Learning*
43 (Withdrawn)
44 Anna Tordai (VU) *On Combining Alignment Techniques*
45 Benedikt Kratz (UvT) *A Model and Language for Business-aware Transactions*
49 Michael Kaisers (UM) *Learning against Learning - Evolutionary dynamics of reinforcement learning algorithms in strategic interactions*
50 Steven van Kervel (TUD) *Ontologogy driven Enterprise Information Systems Engineering*
51 Jeroen de Jong (TUD) *Heuristics in Dynamic Sceduling; a practical framework with a case study in elevator dispatching*

2013

1 Viorel Milea (EUR) *News Analytics for Financial Decision Support*
2 Erietta Liarou (CWI) *MonetDB/DataCell: Leveraging the Column-store Database Technology for Efficient and Scalable Stream Processing*
4 Chetan Yadati (TUD) *Coordinating autonomous planning and scheduling*
5 Dulce Pumareja (UT) *Groupware Requirements Evolutions Patterns*
6 Romulo Goncalves (CWI) *The Data Cyclotron: Juggling Data and Queries for a Data Warehouse Audience*
7 Giel van Lankveld (UvT) *Quantifying Individual Player Differences*
8 Robbert-Jan Merk (VU) *Making enemies: cognitive modeling for opponent agents in fighter pilot simulators*
9 Fabio Gori (RUN) *Metagenomic Data Analysis: Computational Methods and Applications*
10 Jeewanie Jayasinghe Arachchige (UvT) *A Unified Modeling Framework for Service Design.*
11 Evangelos Pournaras (TUD) *Multi-level Reconfigurable Self-organization in Overlay Services*
12 Marian Razavian (VU) *Knowledge-driven Migration to Services*
13 Mohammad Safiri (UT) *Service Tailoring: User-centric creation of integrated IT-based homecare services to support independent living of elderly*
14 Jafar Tanha (UVA) *Ensemble Approaches to Semi-Supervised Learning Learning*
15 Daniel Hennes (UM) *Multiagent Learning - Dynamic Games and Applications*
16 Eric Kok (UU) *Exploring the practical benefits of argumentation in multi-agent deliberation*
17 Koen Kok (VU) *The PowerMatcher: Smart Coordination for the Smart Electricity Grid*
18 Jeroen Janssens (UvT) *Outlier Selection and One-Class Classification*
19 Renze Steenhuizen (TUD) *Coordinated Multi-Agent Planning and Scheduling*
20 Katja Hofmann (UvA) *Fast and Reliable Online Learning to Rank for Information Retrieval*
21 Sander Wubben (UvT) *Text-to-text generation by monolingual machine translation*
22 Tom Claassen (RUN) *Causal Discovery and Logic*
23 Patricio de Alencar Silva (UvT) *Value Activity Monitoring*
24 Haitham Bou Ammar (UM) *Automated Transfer in Reinforcement Learning*
27 Mohammad Huq (UT) *Inference-based Framework Managing Data Provenance*
28 Frans van der Sluis (UT) *When Complexity becomes Interesting: An Inquiry into the Information eXperience*
29 Iwan de Kok (UT) *Listening Heads*
30 Joyce Nakatumba (TUE) *Resource-Aware Business Process Management: Analysis and Support*
31 Dinh Khoa Nguyen (UvT) *Blueprint Model and Language for Engineering Cloud Applications*
32 Kamakshi Rajagopal (OUN) *Networking For Learning; The role of Networking in a Lifelong Learner's Professional Development*
33 Qi Gao (TUD) *User Modeling and Personalization in the Microblogging Sphere*
34 Kien Tjin-Kam-Jet (UT) *Distributed Deep Web Search*
35 Abdallah El Ali (UvA) *Minimal Mobile Human Computer Interaction*
36 Than Lam Hoang (TUe) *Pattern Mining in Data Streams*
37 Dirk Börner (OUN) *Ambient Learning Displays*
39 Joop de Jong (TUD) *A Method for Enterprise Ontology based Design of Enterprise Information Systems*
40 Pim Nijssen (UM) *Monte-Carlo Tree Search for Multi-Player Games*
41 Jochem Liem (UVA) *Supporting the Conceptual Modelling of Dynamic Systems: A Knowledge Engineering Perspective on Qualitative Reasoning*
42 Léon Planken (TUD) *Algorithms for Simple Temporal Reasoning*
43 Marc Bron (UVA) *Exploration and Contextualization through Interaction and Concepts*

2014

1 Nicola Barile (UU) *Studies in Learning Monotone Models from Data*
2 Fiona Tuliyano (RUN) *Combining System Dynamics with a Domain Modeling Method*
3 Sergio Raul Duarte Torres (UT) *Information Retrieval for Children: Search Behavior and Solutions*
4 Hanna Jochmann-Mannak (UT) *Websites for children: search strategies and interface design - Three studies on children's search performance and evaluation*
5 Jurriaan van Reijsen (UU) *Knowledge Perspectives on Advancing Dynamic Capability*
6 Damian Tamburri (VU) *Supporting Networked Software Development*
7 Arya Adriansyah (TUE) *Aligning Observed and Modeled Behavior*
8 Samur Araujo (TUD) *Data Integration over Distributed and Heterogeneous Data Endpoints*
11 Janneke van der Zwaan (TUD) *An Empathic Virtual Buddy for Social Support*
12 Willem van Willigen (VU) *Look Ma, No Hands: Aspects of Autonomous Vehicle Control*
13 Arlette van Wissen (VU) *Agent-Based Support for Behavior Change: Models and Applications in Health and Safety Domains*
14 Yangyang Shi (TUD) *Language Models With Meta-information*
15 Natalya Mogles (VU) *Agent-Based Analysis and Support of Human Functioning in Complex Socio-Technical Systems: Applications in Safety and Healthcare*
16 Krystyna Milian (VU) *Supporting trial recruitment and design by automatically interpreting eligibility criteria*
17 Kathrin Dentler (VU) *Computing healthcare quality indicators automatically: Secondary Use of Patient Data and Semantic Interoperability*
18 Mattijs Ghijsen (UVA) *Methods and Models for the Design and Study of Dynamic Agent Organizations*

19 Vinicius Ramos (TUE) *Adaptive Hypermedia Courses: Qualitative and Quantitative Evaluation and Tool Support*
20 Mena Habib (UT) *Named Entity Extraction and Disambiguation for Informal Text: The Missing Link*
21 Kassidy Clark (TUD) *Negotiation and Monitoring in Open Environments*
22 Marieke Peeters (UU) *Personalized Educational Games - Developing agent-supported scenario-based training*
23 Eleftherios Sidirourgos (UvA/CWI) *Space Efficient Indexes for the Big Data Era*
24 Davide Ceolin (VU) *Trusting Semi-structured Web Data*
25 Martijn Lappenschaar (RUN) *New network models for the analysis of disease interaction*
26 Tim Baarslag (TUD) *What to Bid and When to Stop*
27 Rui Jorge Almeida (EUR) *Conditional Density Models Integrating Fuzzy and Probabilistic Representations of Uncertainty*
29 Jaap Kabbedijk (UU) *Variability in Multi-Tenant Enterprise Software*
30 Peter de Cock (UvT) *Anticipating Criminal Behaviour*
31 Leo van Moergestel (UU) *Agent Technology in Agile Multiparallel Manufacturing and Product Support*
32 Naser Ayat (UvA) *On Entity Resolution in Probabilistic Data*
33 Tesfa Tegegne (RUN) *Service Discovery in eHealth*
34 Christina Manteli (VU) *The Effect of Governance in Global Software Development: Analyzing Transactive Memory Systems.*
35 Joost van Ooijen (UU) *Cognitive Agents in Virtual Worlds: A Middleware Design Approach*
36 Joos Buijs (TUE) *Flexible Evolutionary Algorithms for Mining Structured Process Models*
37 Maral Dadvar (UT) *Experts and Machines United Against Cyberbullying*
38 Danny Plass-Oude Bos (UT) *Making brain-computer interfaces better: improving usability through post-processing.*
39 Jasmina Maric (UvT) *Web Communities, Immigration, and Social Capital*
40 Walter Omona (RUN) *A Framework for Knowledge Management Using ICT in Higher Education*
41 Frederic Hogenboom (EUR) *Automated Detection of Financial Events in News Text*
42 Carsten Eijckhof (CWI/TUD) *Contextual Multidimensional Relevance Models*
43 Kevin Vlaanderen (UU) *Supporting Process Improvement using Method Increments*
44 Paulien Meesters (UvT) *Intelligent Blauw. Met als ondertitel: Intelligence-gestuurde politiezorg in gebiedsgebonden eenheden.*
45 Birgit Schmitz (OUN) *Mobile Games for Learning: A Pattern-Based Approach*
46 Ke Tao (TUD) *Social Web Data Analytics: Relevance, Redundancy, Diversity*
47 Shangsong Liang (UVA) *Fusion and Diversification in Information Retrieval*

2015

1 Niels Netten (UvA) *Machine Learning for Relevance of Information in Crisis Response*
2 Faiza Bukhsh (UvT) *Smart auditing: Innovative Compliance Checking in Customs Controls*
3 Twan van Laarhoven (RUN) *Machine learning for network data*
4 Howard Spoelstra (OUN) *Collaborations in Open Learning Environments*
6 Farideh Heidari (TUD) *Business Process Quality Computation - Computing Non-Functional Requirements to Improve Business Processes*
7 Maria-Hendrike Peetz (UvA) *Time-Aware Online Reputation Analysis*
8 Jie Jiang (TUD) *Organizational Compliance: An agent-based model for designing and evaluating organizational interactions*
9 Randy Klaassen (UT) *HCI Perspectives on Behavior Change Support Systems*
10 Henry Hermans (OUN) *OpenU: design of an integrated system to support lifelong learning*
11 Yongming Luo (TUE) *Designing algorithms for big graph datasets: A study of computing bisimulation and joins*
12 Julie M. Birkholz (VU) *Modi Operandi of Social Network Dynamics: The Effect of Context on Scientific Collaboration Networks*
13 Giuseppe Procaccianti (VU) *Energy-Efficient Software*
14 Bart van Straalen (UT) *A cognitive approach to modeling bad news conversations*
15 Christoph Bösch (UT) *Cryptographically Enforced Search Pattern Hiding*
15 Klaas Andries de Graaf (VU) *Ontology-based Software Architecture Documentation*
17 André van Cleeff (UT) *Physical and Digital Security Mechanisms: Properties, Combinations and Trade-offs*
18 Holger Pirk (CWI) *Waste Not, Want Not! - Managing Relational Data in Asymmetric Memories*
19 Bernardo Tabuenca (OUN) *Ubiquitous Technology for Lifelong Learners*
20 Loïs Vanhée (UU) *Using Culture and Values to Support Flexible Coordination*
23 Luit Gazendam (VU) *Cataloguer Support in Cultural Heritage*
24 Richard Berendsen (UVA) *Finding People, Papers, and Posts: Vertical Search Algorithms and Evaluation*
25 Steven Woudenberg (UU) *Bayesian Tools for Early Disease Detection*
26 Alexander Hogenboom (EUR) *Sentiment Analysis of Text Guided by Semantics and Structure*
27 Sándor Héman (CWI) *Updating compressed colomn stores*
28 Janet Bagorogoza (TiU) *KNOWLEDGE MANAGEMENT AND HIGH PERFORMANCE; The Uganda Financial Institutions Model for HPO*
29 Hendrik Baier (UM) *Monte-Carlo Tree Search Enhancements for One-Player and Two-Player Domains*
30 Kiavash Bahreini (OU) *Real-time Multimodal Emotion Recognition in E-Learning*
31 Yakup Koç (TUD) *On the robustness of Power Grids*
32 Jerome Gard (UL) *Corporate Venture Management in SMEs*
33 Frederik Schadd (TUD) *Ontology Mapping with Auxiliary Resources*
34 Victor de Graaf (UT) *Gesocial Recommender Systems*
35 Jungxao Xu (TUD) *Affective Body Language of Humanoid Robots: Perception and Effects in Human Robot Interaction*

2016

1 Syed Saiden Abbas (RUN) *Recognition of Shapes by Humans and Machines*
2 Michiel Christiaan Meulendijk (UU) *Optimizing medication reviews through decision support: prescribing a better pill to swallow*
3 Maya Sappelli (RUN) *Knowledge Work in Context: User Centered Knowledge Worker Support*
4 Laurens Rietveld (VU) *Publishing and Consuming Linked Data*
5 Evgeny Sherkhonov (UVA) *Expanded Acyclic Queries: Containment and an Application in Explaining Missing Answers*
6 Michel Wilson (TUD) *Robust scheduling in an uncertain environment*
7 Jeroen de Man (VU) *Measuring and modeling negative emotions for virtual training*
8 Matje van de Camp (TiU) *A Link to the Past: Constructing Historical Social Networks from Unstructured Data*
9 Archana Nottamkandath (VU) *Trusting Crowdsourced Information on Cultural Artefacts*
10 George Karafotias (VUA) *Parameter Control for Evolutionary Algorithms*
11 Anne Schuth (UVA) *Search Engines that Learn from Their Users*
12 Max Knobbout (UU) *Logics for Modelling and Verifying Normative Multi-Agent Systems*
13 Nana Baah Gyan (VU) *The Web, Speech Technologies and Rural Development in West Africa - An*
14 Ravi Khadka (UU) *Revisiting Legacy Software System Modernization*
15 Steffen Michels (RUN) *Hybrid Probabilistic Logics - Theoretical Aspects, Algorithms and Experiments*
16 Guangliang Li (UVA) *Socially Intelligent Autonomous Agents that Learn from Human Reward*
17 Berend Weel (VU) *Towards Embodied Evolution of Robot Organisms*
18 Albert Meroño Peñuela *Refining Statistical Data on the Web*
19 Julia Efremova (Tu/e) *Mining Social Structures from Genealogical Data*
20 Daan Odijk (UVA) *Context & Semantics in News & Web Search*
21 Alejandro Moreno Célleri (UT) *From Traditional to Interactive Playspaces: Automatic Analysis of Player Behavior in the Interactive Tag Playground*
22 Grace Lewis (VU) *Software Architecture Strategies for Cyber-Foraging Systems*
23 Fei Cai (UVA) *Query Auto Completion in Information Retrieval*
24 Brend Wanders (UT) *Repurposing and Probabilistic Integration of Data; An Iterative and data model independent approach*
25 Julia Kiseleva (TU/e) *Using Contextual Information to Understand Searching and Browsing Behavior*
26 Dilhan Thilakarathne (VU) *In or Out of Control: Exploring Computational Models to Study the Role of Human Awareness and Control in Behavioural Choices, with Applications in Aviation and Energy Management Domains*
27 Wen Li (TUD) *Understanding Geo-spatial Information on Social Media*
28 Mingxin Zhang (TUD) *Large-scale Agent-based Social Simulation - A study on epidemic prediction and control*
29 Nicolas Höning (TUD) *Peak reduction in decentralised electricity systems -Markets and prices for flexible planning*
30 Ruud Mattheij (UvT) *The Eyes Have It*
31 Mohammad Khelghati (UT) *Deep web content monitoring*
32 Eelco Vriezekolk (UT) *Assessing Telecommunication Service Availability Risks for Crisis Organisations*
33 Peter Bloem (UVA) *Single Sample Statistics, exercises in learning from just one example*
34 Dennis Schunselaar (TUE) *Configurable Process Trees: Elicitation, Analysis, and Enactment*
35 Zhaochun Ren (UVA) *Monitoring Social Media: Summarization, Classification and Recommendation*
36 Daphne Karreman (UT) *Beyond R2D2: The design of nonverbal interaction behavior optimized for robot-specific morphologies*
37 Giovanni Sileno (UvA) *Aligning Law and Action - a conceptual and computational inquiry*
38 Andrea Minuto (UT) *MATERIALS THAT MATTER - Smart Materials meet Art & Interaction Design*
39 Merijn Bruijnes (UT) *Believable Suspect Agents; Response and Interpersonal Style Selection for an Artificial Suspect*
40 Christian Detweiler (TUD) *Accounting for Values in Design*
41 Thomas King (TUD) *Governing Governance: A Formal Framework for Analysing Institutional Design and Enactment Governance*
42 Spyros Martzoukos (UVA) *Combinatorial and Compositional Aspects of Bilingual Aligned Corpora*