Text-to-text generation by monolingual machine translation

Wubben, S.

Publication date:
2013

Text-to-Text Generation by Monolingual Machine Translation

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan Tilburg University
op gezag van de rector magnificus,
prof. dr. Ph. Eijlander,
in het openbaar te verdedigen ten overstaan van een
door het college van promoties aangewezen commissie
in de aula van de Universiteit
op woensdag 5 juni 2013 om 16:15 uur

door

Sander Wubben,
geboren op 23 juli 1980 te Epe

PROMOTORES:
Prof. dr. E.J. Krahmer
Prof. dr. A.J.P. van den Bosch
Prof. dr. H.C. Bunt

COMMITTEE:
Prof. dr. W.M.P. Daelemans
Dr. W.B. Dolan
Dr. K. Filippova
Dr. E.C. Marsi
Dr. A. Siddharthan
Dr. M. Theune


SIKS Dissertation Series No. 2013-21

The research in this thesis had been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.


TiCC PhD Dissertation Series No. 26

ISBN/EAN: 978-94-6203-349-8
Printed by CPI Koninklijke Wöhrmann
Cover design by Sander Wubben
This document was typeset using the typographical look-and-feel `classic thesis`

The Revolution will be complete when the language is perfect.

— George Orwell, 1984

PREFACE

Finishing my PhD thesis I've come to realise that although I started my research in 2008, the actual seeds were in fact planted much earlier. I remember vividly the computer that my parents owned when I was around twelve, and on which I spend countless hours fiddling about. It was a cream coloured 8086 XT PC with an impressive amber monochrome monitor. This machine opened up an entire new world to me. I quickly discovered the fun of writing small programs in BASIC. One of my biggest accomplishments was writing a program that could generate a thousand unique sentences by combining a limited number of sentence parts. The challenge was in generating sentences that were as absurd and profane as possible. Twenty years later, I have finished my thesis on text-to-text generation and although the challenges are somewhat different, the fun in automatically generating text remains the same.

Getting to this point has been quite a journey and there are many people who have helped me along the way. First of all, I would like to sincerely thank my supervisors. I am lucky enough to have three of them and I am very grateful for their dedication, guidance, patience and the amount of freedom they gave me in pursuing my research. They have often gone far beyond what can realistically be expected of any supervisor.

Before becoming my PhD supervisor Antal van den Bosch supervised my Master's thesis. Even before that, he was instrumental in getting me involved in computational linguistics. I always thoroughly enjoyed meeting with Antal, as our discussions were always interesting and he was more than willing to listen to any ideas I had and discuss them, even if they were on important topics such as science fiction, games or gerbils. One of the fascinating aspects of Antal is that you can send him an email in the middle of the night and still receive a reply within five minutes. There have been numerous occasions where he kept giving detailed feedback while I was finishing a paper the night before a deadline.

If Antal was my main supervisor in the first half of my project, then Emiel Krahmer became my main supervisor in the last half. I am very grateful for his never-ending positivism and enthusiasm. At times results and reviews could seem depressing, but Emiel always managed to cheer me up and helped me focus on the bigger picture. His organisational skills really helped me finish my thesis in a reasonable time. I also greatly appreciated his opinion on music ranging from Dr. Dre to the terrors of folk music. His sense of humour is also something that is remarkable. I'm never quite sure if I actually get his jokes or not.

When I started my project Harry Bunt quickly got me involved in organising the IWCS conference, which was a very valuable experience. Compiling the IWCS proceedings has made compiling this thesis a breeze. Harry always had a fresh look on my research and often pointed out things that hadn't even occurred to me. The comments he made on my texts helped to improve them greatly. His stories and tips on travel, being it to Schiermonnikoog or to Jeju island, were also very enjoyable and helpful.

I'm greatly indebted to my excellent committee. Many thanks for all the kind words and valuable comments that helped make this thesis a lot better. Thanks are also due to Mike Kestemont for his advise on Middle Dutch and to Martin Reynaert for proofreading the manuscript. I thank Zhemin Zhu and Kristian Woodsend for sharing their data with me for the experiments conducted in Chapter 3.

During my PhD project, I very much enjoyed the positive atmosphere within the Induction of Linguistic Knowledge research group which resulted in coffee experimentation, ILK barbies and Guitar Hero sessions. I'd like to thank all the people who made this possible. Thanks also go out to the members of the føøf pubquiz team for their excellent trivia knowledge and to my roommates for sharing an office, sharing ideas and for putting up with my dead plant collection. This thesis can not be complete without a mention of Hans Paijmans. Paai supervised my Bachelor's project and managed to make it a unique experience. That experience made me very excited about doing research and played a great part in my choice of pursuing a career in academia. The last year involved quite some teaching and I'm indebted to the colleagues I collaborated with for making this go very smoothly. Thank you Menno, Ruud, Pieter, Suleman and Grzegorz!

Finally, thanks remain to the most important people in my life. I would like to express my sincere gratitude to my parents for always believing in me. I would not be where I am today without their support. Each journey needs detours and I'm happy that my friends were there to provide some distractions. Thanks specifically to Niels and Paul for numerous great evenings full of philosophical and not so philosophical discussions in full or empty bars

and to Jurrit for hauling me up rocks and mountains. Many many thanks to my lovely wife Marianne. She has always provided me with lots of love, fun, support and understanding. It has been quite a journey, and I'm certain that many more adventures await us!

Tilburg, May 2013

CONTENTS

1

INTRODUCTION

1.1 TEXT-TO-TEXT GENERATION

Text-to-text generation is generally defined as automatically producing a target text from a source text in the same language. It is part of the larger field of Natural Language Generation (NLG), which is the part of artificial intelligence that deals with the natural language processing task of generating natural language (McDonald and Pustejovsky, 1985; Paris et al., 1991; Bateman, 1997; Ratnaparkhi, 2000; Reiter and Dale, 2000). Traditionally NLG was mainly concerned with generating natural language from non-linguistic data, such as knowledge databases or logical forms (Reiter and Dale, 2000). One example is the automatic generation of weather forecast reports from weather data (Goldberg et al., 1994; Reiter and Dale, 1997; Belz, 2008). In the last decade the work in NLG has shifted gradually towards more data-oriented approaches (Krahmer and Theune, 2010). With the availability of more and more textual data for these approaches, partly due to the growth of the web, text-to-text generation has been able to flourish and is considered an increasingly important part of NLG.

Nowadays, many popular natural language applications are in fact text-to-text applications. Question answering (QA), for instance, can be applied to collections of documents instead of databases. QA is an application that takes a question as input and generates an answer in natural language that should contain the answer to the question posed (Voorhees, 2001; Ravichan-

dran and Hovy, 2002). For example, a user can enter the query "Who is the author of the Lord of the Rings?" and expect the answer "J.R.R. Tolkien". The QA system will retrieve the answer to the question by querying a document collection, matching the question and returning the result in natural language. Another application is automatic summarization (Corston-Oliver, 2001; Mani, 2001; Neto et al., 2002; Daume and Marcu, 2005). In automatic summarization, texts are shortened automatically by selecting only the most relevant sentences and removing redundant words. These are two examples of applications that process natural language input and turn it into natural language output, and can be seen as application domains of text-to-text generation techniques.

One of the main challenges in text-to-text generation is generating well-formed output. The sentences that a text-to-text application produces should be grammatical (often called fluency) and should have a meaningful relation with its input (such as an entailment relation), depending on the application (often called adequacy). Besides that, additional constraints can be introduced: for example, in summarization the output text should be shorter than the input text, in Question Answering the output should contain the information that the question asks for.

Sentential paraphrase generation, sentence simplification, sentence compression and sentence fusion are all text-to-text generation techniques operating on the sentence level (Dras, 1997). This means that text generation is in these cases restricted to sentence generation. This also means that it is a challenge to model text features such as the use of anaphors and context are often not modeled adequately. However, Siddharthan (2006) describes a method to preserve cohesive relations during the simplification process. Sentential text-to-text generation techniques can be helpful for various NLP applications, such as Information Retrieval and Question Answering (for instance, by query expansion (McKeown, 1979; Anick and Tipirneni, 1999; Ravichandran and Hovy, 2002; Riezler et al., 2007)). If a question Answering (QA) system is unable to find an answer for a question, the system can benefit from paraphrasing or simplifying the question. For example, when the question is : "Who *is the author of* the Lord of the Rings?" And the document collection that the QA system queries only contains "J.R.R Tolkien *wrote* the Lord of the Rings" paraphrasing "is the author of" into "wrote" would greatly help the system. Sentential paraphrase generation can also help machine translation (MT). A sentence that cannot be translated properly by a system because parts of it are not known by the MT system can be paraphrased into a sentence the system can translate containing only words and phrases that are. Sentence compression and sentence fusion can help create a coherent summary of a text or of multiple texts by removing non-

essential parts of a sentence or by combining two or more related sentences into one sentence containing the important bits, thereby improving automatic summarization and multi-document summarization (Barzilay et al., 1999; Barzilay and Elhadad, 2003; Marsi and Krahmer, 2005; Filippova and Strube, 2008; Hendrickx et al., 2009). Paraphrase generation can also be valuable for the automatic evaluation of machine translation or automatic summarization. Typically automatic evaluation of these tasks relies on matching the output of the system to a set of human produced reference translations or summaries. The coverage of these references can be increased by paraphrasing them to create more diverse references (Zhou et al., 2006; Kauchak and Barzilay, 2006; Madnani et al., 2007; Snover et al., 2010).

Since the monolingual text-to-text generation field is relatively new, no established methods exist yet. The challenges faced in the research area of machine translation are however similar to the challenges in monolingual text-to-text generation. Both disciplines are for example involved in generating grammatically correct output and in generating output that is meaningfully related to the input of the system. An interesting venue for research is then the application of established methods in machine translation to diverse text-to-text generation tasks. This bring us to the first research question we aim to answer in this thesis, which is:

1. How can a statistical machine translation model be applied to a collection of monolingual text-to-text generation tasks?

One of the main challenges in text-to-text-generation is acquiring good quality data. In statistical machine translation this can be done relatively easily, for instance by training the model on examples of translations into the desired languages of the proceedings of the European Parliament (Koehn, 2005). For text-to-text generation this is typically less straightforward. The nature of the data is dependent on the task it is used for. For paraphrase generation for instance, we would need a similar parallel corpus as used in statistical machine translation. But instead of aligning sentences and their translations, we now need to have sentences aligned with their paraphrases. Large paraphrase corpora are hard to come by. The same is true for corpora for sentence simplification and sentence compression. The corpora that are available are often relatively small and most of the data is English. The second research question we pose is then:

2. How can good parallel monolingual corpora be created?

In addition to the challenge of the acquisition of data, there is the issue of evaluation. One factor in making successful text-to-text applications is a

proper evaluation methodology. In Machine Translation the methodology of automatic evaluation is well established (Papineni et al., 2002). The BLEU metric and other similar metrics compare the machine translation output to some reference translations and measure to what degree they match by counting n-gram overlaps. Papineni et al. (2002) have demonstrated that BLEU correlates well with human judgements[1]. Our third research question, then, is if a similar evaluation methodology can be applied to text-to-text generation, where previously mostly human judgements have been used to assess the output of NLG systems:

3. To what extent can text-to-text generation be evaluated automatically?

In the remainder of this chapter, we will first discuss Statistical Machine Translation and its application to monolingual text-to-text generation problems, and then we will discuss the subareas we will cover, namely paraphrase generation, sentence simplification, sentence compression, and language transformation.

1.2 STATISTICAL MACHINE TRANSLATION

Statistical Machine Translation (SMT) is the process where target translations are generated from a source text on the basis of statistical models whose parameters are derived from large bilingual text corpora aligned on the sentence level (Brown et al., 1990, 1993; Och et al., 1999; Och and Ney, 2000b). Large parallel corpora, such as the multilingual proceedings of the European Parliament (Europarl), are available for many language pairs. The SMT approach regards the translation process as a stochastic optimization problem. A learning algorithm can be applied to the bilingual corpus which models the translation process, by finding alignments between words or phrases in the two languages. With the resulting probabilistic model, previously unseen sentences in the source language can then be automatically translated in the target language by a decoder.

Let us assume that we wish to translate a sentence F in one language, whose words are $f_1, f_2, ..., f_n$ to a sentence E in another language. If we wish to find the optimal translation $\hat{E} = e_1, e_2, ..., e_m$, we are searching for the most likely translation. This means we are searching for the sentence E for which the probability of $P(E|F)$ is the highest. Using Bayes' rules we can rewrite this as

1 BLEU has received criticism as well by for instance Callison-Burch et al. (2006b), who argue human evaluation should be used when comparing MT systems

$$\begin{aligned}\hat{E} &= \operatorname{argmax}_E P(E|F) \\ &= \operatorname{argmax}_E \frac{P(F|E)P(E)}{P(F)} \\ &= \operatorname{argmax}_E P(F|E)P(E)\end{aligned}$$

Here, $P(F|E)$ models the quality of the phrase translations (adequacy) and is represented by the translation model. $P(E)$ models the fluency of the resulting sentence and can be modeled using a language model. The language model is typically an n-gram model trained on monolingual data, such as SRILM (Stolcke, 2002).

Instead of operating just on the word level, most SMT systems nowadays operate on the phrase level, where phrases are sequences of words (Zens et al., 2002; Koehn et al., 2003). Phrase-based machine translation (PBMT) attempts to translate each phrase $\bar{f}_i$ from sentence F in one language into a phrase $\bar{e}_i$ that will be part of sentence E in another language. This is denoted by the probability distribution $\phi(\bar{f}_i|\bar{e}_i)$ After the translation process there is an optional reordering process called distortion, which accounts for word order variation between languages. The further away from its initial position a translated phrase is placed, the lower its distortion probability will be. Distortion is described by $d(a_i - b_{i-1})$ where a_i is the start position of the translated phrase $\bar{e}_i$ and b_{i-1} is the end position of the translation of the phrase $\bar{e}_{i-1}$. When we focus only on translation and leave the language model temporarily out of the picture, the translation process can be described by

$$P(F|E) = \sum_{i=1}^{I} \phi(\bar{f}_i|\bar{e}_i) d(a_i - b_{i-1})$$

There are two phases which can be distinguished in the translation pipeline: alignment and decoding. The alignment procedure can start with word alignment using IBM models and a probabilistic Hidden Markov Model (HMM) alignment algorithm (Och and Ney, 2000a). An example of a word alignment matrix can be seen in Figure 1.1. These word alignments are then combined into phrase alignments by a process called symmetrizing. The alignment process produces a phrase table with aligned phrases and scores indicating the probability of the alignment. An example from a phrase table for Dutch - English is given in Table 1.1. The decoder tries to translate a source sentence by finding the target sentence that optimizes the translation and language model probabilities. This is typically a search problem, and most decoders

use a variant of $A*$ search, a heuristically informed search algorithm (Koehn et al., 2003).

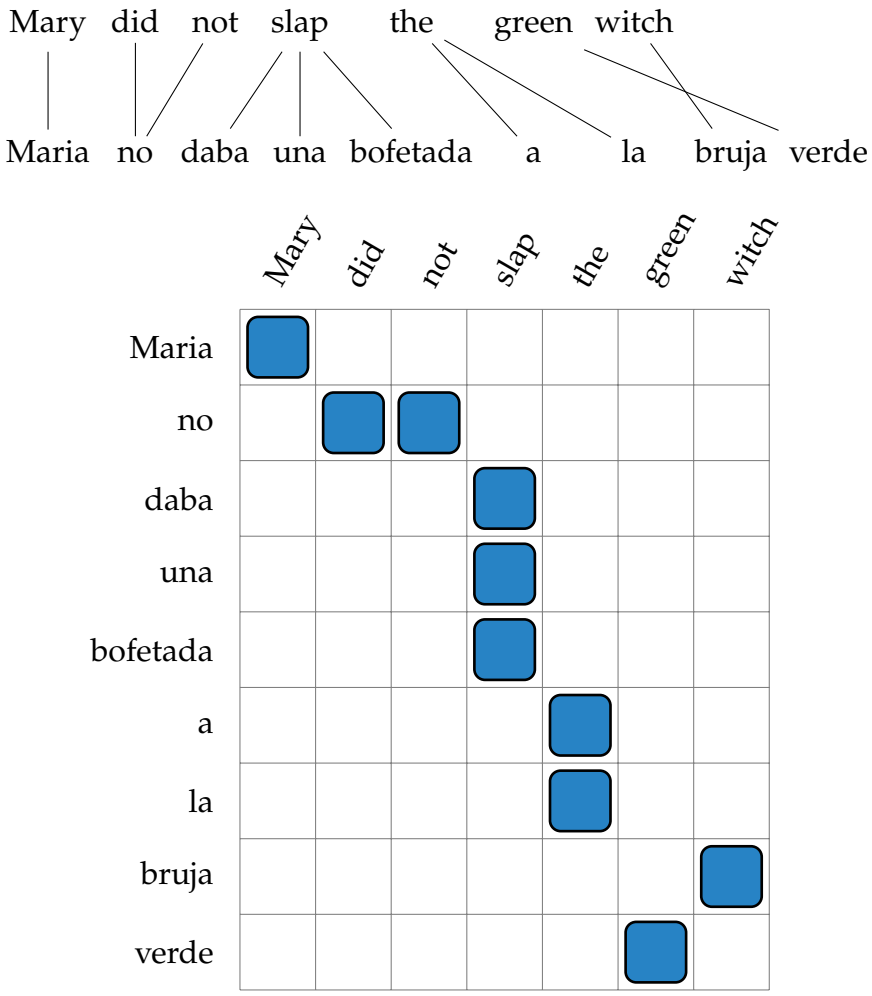


Figure 1.1: An example of word alignments between an English and a Spanish sentence

1.3 STATISTICAL MACHINE TRANSLATION FOR TEXT-TO-TEXT GENERATION

Monolingual text-to-text generation can be seen as a translation task within a language. One of the requirements for such an approach is a monolingual parallel corpus. F and E are now sentences in the same language, and the output sentence E has to meet certain constraints. In the case of paraphrasing, E should be different in form from F but roughly equal in meaning. In the case of compression, E should be shorter than F, but should still contain

source phrase	target phrase	$\phi(f|e)$	$\phi(e|f)$
banaan is	bananas have	0.5	1
banaan	banana of	1	0.0833333
banaan	banana	0.0416667	0.416667
banaan	bananas	0.0145278	0.5

Table 1.1: Sample from a Dutch - English phrase table, showing phrases with alignment scores

the most important information, and in case of simplification it should be simpler than F. If we view these tasks as translation tasks, we have to collect monolingual corpora for these tasks. These corpora are not as readily available as bilingual corpora, and when they are available they are significantly smaller than most bilingual corpora. We will discuss several viable approaches to collecting data in order to construct corpora for these tasks.

One way of constructing a monolingual parallel corpus is the use of multiple translations of the same source texts. Different translators may have different ways of translating the information in a source text, meaning the different output sentences are in fact paraphrases. They contain the same information in different wordings. Barzilay and McKeown (2001) constructed a corpus containing multiple English translations of five classic novels including Madame Bovary and 20,000 Leagues Under the Sea, and Marsi and Krahmer (2007) did this for Dutch. Below is an example of two translations of a sentence found in different translated versions of Madame Bovary:

(1.1)
1. Emma burst into tears and he tried to comfort her, saying things to make her smile.
2. Emma cried, and he tried to console her, adorning his words with puns.

One issue that arises when using multiple translations is the problem of sentence alignment. Sometimes sentences shift position or sentences are fused or split. This means that the information contained in one sentence in one translation can be spread over multiple sentences in another translation. Barzilay and McKeown (2001) use the sentence alignment techniques described by Gale and Church (1993) as a first step to construct their corpus.

Pang et al. (2003) used multiple translations of Chinese news articles to obtain a corpus of paraphrases. They obtained these translations from the

Multiple-Translation Chinese Corpus, originally created to be used with the BLEU Machine Translation evaluation metric (Papineni et al., 2002). Sentences from these translations were manually aligned.

Quirk et al. (2004) were the first to try collecting monolingual parallel corpora for paraphrasing by using clusters of news articles that cover the same event. Because the articles in each cluster do not always report the same information they are not strictly parallel texts, but comparable texts. They are, however, likely to contain words, phrases and sentences with similar meaning, because they report on the same event. From each cluster, Quirk and colleagues selected comparable sentences by using a string edit distance heuristic proposed by Dolan et al. (2004). They used the resulting corpus for monolingual machine translation. Below are two sentential paraphrases from their corpus:

(1.2)
1. Dzeirkhanov said 36 people were injured and that four people, including a child, had been hospitalized.
2. Of the 36 wounded, four people including one child, were hospitalized, Dzheirkhanov said.

This example contains rewording, rephrasing, deletion, insertion and reordering. These are all possible operations in monolingual text-to-text generation.

authors	source	n aligned sentences
Barzilay and McKeown (2001)	Translations of books	26,201
Pang et al. (2003)	Translations of news articles	109,230
Quirk et al. (2004)	Clusters of news articles	153,403

Table 1.2: Number of aligned paraphrases of various corpora derived from monolingual parallel or comparable corpora

Table 1.2 lists a collection of text-to-text corpora containing paraphrases along with their size. If we take into account that statistical machine translation systems can easily use models of several million aligned sentences, we can conclude that the existing monolingual text-to-text corpora are not particularly large.

One way of obtaining sufficiently large corpora for text-to-text generation tailored towards paraphrasing is the use of bilingual parallel corpora. The advantage is that the abundance of bilingual corpora can be exploited for

monolingual tasks in this way. The extraction of paraphrases by using a pivot approach on bilingual corpora was proposed by Bannard and Burch (2005). English paraphrases are obtained by pivoting through foreign language phrases, given a phrase table containing English phrases aligned to foreign phrases. An example of this approach can be seen in Figure 1.2

Generally, multiple paraphrases can be extracted for each source phrase. These can be ranked by using translation model probabilities:

$$\begin{aligned} P(E_2|E_1) &= \sum_F P(E_2, F|E_1) \\ &= \sum_F P(E_2|F, E_1)P(F|E_1) \\ &\approx \sum_F P(E_2|F)P(F|E_1) \end{aligned}$$

Here, E_1 is the source phrase and E_2 the target phrase in the desired language and F is the phrase in a foreign language.

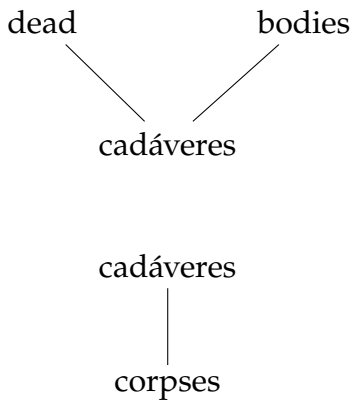


Figure 1.2: By pivoting over a Spanish translation, the paraphrasing phrase "dead bodies" for "corpses" can be discovered

The pivot approach has since been extended and improved by using syntactic information (Chiang et al., 2005; Madnani et al., 2007; Callison-Burch, 2008; Zhao et al., 2009) and extended to be able to handle sentential paraphrases (Ganitkevitch et al., 2011). Although the pivot approach allows us to collect larger paraphrase corpora, there are some drawbacks. First of all, the extracted paraphrase pairs are quite restricted. Pivoted phrases will for instance not likely contain re-orderings. Using syntactic information to get higher quality paraphrases will probably restrict the extracted paraphrases even more. Another drawback is that this approach is specifically tailored

towards paraphrasing and it will therefore probably not be easy to adapt this approach to related tasks such as sentence simplification.

We have discussed mainly approaches concerning paraphrases, but other areas have also received some attention. Zhu et al. (2010) examine the use of paired documents in English Wikipedia and Simple English Wikipedia for sentence simplification. Simple English Wikipedia is an encyclopedia written in simple English mainly for people that have difficulty understanding the regular English Wikipedia, such as children, second language learners, and people with comprehension difficulties such as dyslexia. Zhu et al. (2010) paired sentences from both versions of Wikipedia to build their parallel corpus. (Cohn and Lapata, 2007) constructed a corpus of manual compressions from text from news broadcasts. (Vandeghinste and Tjong Kim Sang, 2004) describe the collection of a parallel corpus of television program transcripts and subtitles in order to perform sentence compression for subtitle generation. They do this for Dutch. In general, it is fair to say, however, that suitable large data-collections of this kind are still thin on the ground.

1.4 THIS THESIS

In this thesis we discuss four variants of monolingual text-to-text generation, which we briefly characterize in the rest of this section. In Chapter 2 we discuss sentential paraphrase generation. Sentential paraphrase generation is the generation of a target sentence that is different in structure from the source sentence, but carries approximately the same meaning. In Chapter 3 we investigate sentence simplification. Sentence simplification is the process of generating a target sentence that is easier to understand than the source sentence, while still largely retaining the same meaning. In Chapter 4 we discuss sentence compression: generating a shorter target sentence that still expresses the most important information from the source sentence. In Chapter 5 we discuss monolingual language transformation. We define this as the process of transforming a sentence from one diachronic variant of a language to a sentence in another diachronic variant of that language. In the final chapter we discuss the results from the previous four chapters and we elaborate on the general conclusions that we draw.

1.4.1 *Paraphrase generation*

Paraphrasing can generally be defined in terms of semantic equivalence. In a strict sense paraphrasing is a form of mutual entailment: sentence A and sentence B are paraphrases if sentence A entails sentence B and sentence B

entails sentence A. But generally a paraphrase is a reformulation of a text into another text that still contains similar semantic content to the original text. Paraphrasing can occur at several levels: individual words can be replaced by their synonyms, but also by more general or more specific words. On a higher level is phrasal paraphrasing, where phrases are replaced by semantically similar phrases. These can be syntactic phrases but also patterns of word sequences. If an entire sentence is rephrased into another sentence that has the same semantic content we speak about a sentential paraphrase (Madnani and Dorr, 2010). An example of a sentential paraphrase is below.

(1.3) 1. The explosion injured twelve people.

2. A dozen persons were wounded by the blast.

In this example we see word substitution using synonyms ("blast" replacing "explosion", "persons" replacing "people") but also phrase replacement ("twelve" is paraphrased into "a dozen") and syntactic rewriting; while the first sentence is active, the second sentence is passive. We have seen that a factor of consideration when paraphrasing automatically with monolingual parallel corpora is the amount of data available. An alternative is to use non-parallel monolingual text and using distributional similarity (Lin and Pantel, 2001b; Bhagat and Ravichandran, 2008). Harris (1954) was the first to propose that a language possesses a distributional structure: words or phrases occurring in the same distribution of contexts are likely to have similar meanings. Lin and Pantel (2001b) created a corpus of sentences from newspaper texts for which they created dependency parses. They measure distributional similarity over paths found in these dependency trees. Ultimately, they induce inference rules that look like this:

$$\text{X found answer to Y} \Leftrightarrow \text{X solved Y}$$
$$\text{X caused Y} \Leftrightarrow \text{Y is blamed on X}$$

Although the amount of data available for this approach is vast, distributional similarity leads to much more noise in the acquired paraphrase patterns. Because the patterns are quite generic, often not only paraphrases are found, but also other related phrases such as hyponyms, co-hyponyms, hypernyms and even antonyms. In addition to that, this approach assumes a dependency parser, which might not be available for every language.

We are interested in the large scale collection of direct paraphrases to be used in a monolingual machine translation system. In Chapter 2, we present

our novel approach to sentential paraphrase generation. We will start with the description of the methodology to collect the training data for this approach. We propose using a news aggregator website to collect news headlines. We argue that headlines are a good source for paraphrasing, because editors and journalists actively rephrase sentences describing a news event in order to produce a unique and compelling headline. Furthermore, these headlines can be collected in large quantities for multiple languages. We will also discuss our proposal for a new sentential paraphrase generation model, which is a phrase-based machine translation approach with re-ranking of the output (PBMT-R), which we argue can easily be used for a variety of text-to-text generation tasks. Finally, we discuss the evaluation of this approach using automatic measures and test subjects.

1.4.2 *Sentence simplification*

Sentence simplification can be defined as the process of producing a version of a sentence that is easier to understand by applying some edit operations to that sentence, while still preserving the semantic content of the original sentence. These operations can include changing some of the lexical material or grammatical structure of that sentence, or deleting difficult superfluous words. This is essentially a variant of paraphrasing, with the additional constraint that the new sentence should be easier to understand than the original. Below is an example from the simplification dataset collected by Zhu et al. (2010):

(1.4)
1. ORIGINAL: However, the bulk of the river flows through tropical rainforest, where there are few roads and even fewer cities, so there is no need for crossings.
2. SIMPLIFICATION: For most of its course, the river flows through tropical rainforest, there are very few roads and cities.

In this example we see that phrases may be paraphrased ("the bulk of the river", "for most of its course"), and can even be deleted ("so there is no need for crossings"). Our aim is to apply the monolingual SMT text-to-text generation approach to sentence simplification by feeding the model simplification data. Simplification data can be harvested from Simple Wikipedia, which is a simplified encyclopedic website, aimed at cognitively impaired readers, second language learners and children. Sentences from the Simple Wikipedia articles can be aligned to their equivalents from the regular English Wikipedia to construct a simplification corpus (Zhu et al., 2010; Woodsend and Lapata, 2011; Coster and Kauchak, 2011). In Chapter 3 we use such

a corpus to generate simplifications and compare to similar approaches. We evaluate the output of these approaches both automatically and by using human judges. Additionally we investigate the feasibility of using automatic readability metrics to evaluate sentence simplification.

1.4.3 *Sentence compression*

Sentence compression is the act of producing a summary of a sentence so that the new sentence is shorter in length than the original sentence. The length of the sentence is reduced by keeping only the most important information contained in the sentence. In this way we summarize, but not on the document but rather on the sentence level. This particular form of compression is useful for applications such as subtitle generation, but also as a preprocessing step to improve summarization applications. The most straightforward manner in which we can compress a sentence is to delete a subset of its words, which can be seen as a variant of extractive summarization (Knight and Marcu, 2002; Turner and Charniak, 2005; McDonald, 2006; Cohn and Lapata, 2007; Galley and McKeown, 2007). An example from the broadcast compression corpus by Cohn and Lapata (2007) is below. This corpus contains manual extractive compressions of sentences from news articles.

(1.5)
1. ORIGINAL: It not only gets you out of that hot kitchen, but according to the Barbecue Industry Association, 91 percent of folks say that they like to cook outdoors because they love the taste of grilled foods.
2. COMPRESSION: It not only gets you out, but according to the Barbecue Industry Association, 91 percent of folks love grilled foods.

Another approach is to rephrase the sentence into a shorter sentence, allowing also paraphrasing operations. This approach has received considerably less attention (Cohn et al., 2008; Zhao et al., 2009). An example of abstractive compression from the abstractive corpus by Cohn et al. (2008) can be observed below.

(1.6)
1. ORIGINAL: Snow, high winds and bitter disagreement yesterday further hampered attempts to tame Mount Etna, which is threatening to overrun the Sicilian town of Zafferana with millions of tons of volcanic lava.

2. COMPRESSION: Bad weather and disagreement yesterday further hampered attempts to tame Mount Etna, which is threatening the Sicilian town of Zafferana.

We are interested in applying the monolingual text-to-text SMT model to the task of generating abstractive compressions of sentences. In Chapter 4 we describe a novel memory-based sentence compression model which performs extractive compression. We describe how we combine this model with the SMT model. We compare this hybrid approach to a purely extractive approach and we let human judges evaluate the output of our systems. Again, we also perform automatic evaluation.

1.4.4 *Language transformation*

Language transformation can be defined as the process of translating between diachronically distinct language variants, such as the translation of Middle English texts to Modern English. Below is an example from the Canterbury Tales:

(1.7)
1. ORIGINAL: A MONK ther was, a fair for the maistrie,
   An outridere, that lovede venerie,
2. TRANSFORMATION: A MONK there was, one of the finest sort,
   An outrider; hunting was his sport;

The first sentence from the example is the Middle English text, the bottom sentence is the modern English transformation. This is again a task that might be tackled by using the monolingual SMT approach. However, for these tasks data is typically sparse. In general, the older the language variant is, the less data there are available. Another point to take into account is that the modern translations often are not literal, for instance because of the additional constraint of a rhyming scheme. On the other hand, the example above demonstrates that the different variants actually exhibit a certain amount of character overlap. In Chapter 5 we describe methods to use character overlap to boost the SMT model. One method we use is introducing a preprocessing step before we use monolingual SMT: we use a dynamic programming approach that finds alignments of phrases based on character overlap. Another approach is to use character bigrams in the translation process instead of words. We compare these approaches to a standard Monolingual SMT approach and let human judges evaluate the output. Additionally we perform automatic evaluation.

1.4.5 *General discussion and conclusion*

In Chapter 6 we draw general conclusions on the results of applying our models to the various text-to-text generation tasks: paraphrase generation, sentence simplification, sentence compression, and language transformation. We will discuss the advantages and disadvantages of using monolingual SMT for these tasks and show where this approach works and where it does not. Additionally, we will discuss our data collection method for paraphrase generation. Finally, we discuss our manual and automatic evaluation methodology.

2

PARAPHRASE GENERATION

In this chapter we investigate the automatic generation of paraphrases by using machine translation techniques. Three contributions we make are the construction of a sufficiently large paraphrase corpus, a re-ranking heuristic to use machine translation for paraphrase generation and a proper evaluation methodology. A large parallel corpus is constructed by aligning clustered headlines that are crawled from a news aggregator site. To generate sentential paraphrases we use a standard phrase-based machine translation (PBMT) framework modified with a re-ranking component (henceforth PBMT-R). We demonstrate this approach for Dutch and English and evaluate by using human judgements collected from 76 participants. The judgments are compared to two automatic machine translation evaluation metrics. We observe that as the paraphrases deviate more from the source sentence, the performance of the PBMT-R system degrades less than that of the word substitution baseline system.

THIS CHAPTER IS BASED ON: Wubben, S., van den Bosch, A.P.J., & Krahmer, E.J. *Creating and using large monolingual parallel corpora for sentential paraphrase generation* (under revision for journal publication)

Earlier versions of this work were presented at: 12th European Workshop on Natural Language Generation, 20th Computational Linguistics in the Netherlands, 6th International Language Generation Conference

2.1 INTRODUCTION

Paraphrasing can be defined as transforming a word, phrase, sentence orl onger text segment in a language from its original surface form to an alternative surface form in the same language that still expresses approximately the same semantic content as the original. An increasing number of Natural Language Processing (NLP) applications rely on the automatic extraction or generation of semantically similar units. This helps these applications broaden their coverage and generally improves their performance. Aside from an interest in using word substitution methods that rely on semantic lexical resources, research in phrasal and sentential paraphrase generation has become more and more popular.

The use of paraphrase generation has been demonstrated to be valuable for question answering (Lin and Pantel, 2001a; Riezler et al., 2007), machine translation (Callison-Burch et al., 2006a; Marton et al., 2009) and the evaluation thereof (Russo-Lassner et al., 2006; Kauchak and Barzilay, 2006; Zhou et al., 2006; Pado et al., 2009). Adding certain constraints to paraphrasing allows for additional useful applications. When the constraint is specified that a paraphrase should be shorter than the input text, paraphrasing can be used for sentence compression (Knight and Marcu, 2002; Barzilay and Lee, 2003). Another specific task that can be approached this way is text simplification, to convert for example medical terms into layperson's English (Elhadad and Sutaria, 2007; Deléger et al., 2009), or for subtitle generation (Daelemans et al., 2004).

Two important problems arise when developing a system that learns to generate paraphrases automatically from examples, namely how to obtain a sufficient number of examples to train the system on, and how to evaluate properly. We present a paraphrase corpus composed of data crawled from Google News to create a parallel corpus, and a standard PBMT framework modified with a re-ranking component (PBMT-R) to learn phrase alignments and generate paraphrases. We demonstrate this approach on Dutch and English and perform an extensive evaluation using human judgements collected from 76 participants, as well as two automatic machine translation evaluation metrics. Our approach can easily be adapted to other languages.

2.1.1 *Phrase-based machine translation (PBMT) for paraphrasing*

Sentential paraphrase generation can be approached as a monolingual machine translation task, where the source and target languages are the same

(Quirk et al., 2004; Bannard and Burch, 2005; Callison-Burch, 2008; Zhao et al., 2009; Wubben et al., 2010) and where the output should be different in form from the input but similar in meaning. Statistical machine translation (SMT) typically makes use of large parallel corpora to train a model on. These corpora need to be aligned at the sentence level. Large parallel corpora, such as the multilingual proceedings of the European Parliament (Europarl), are readily available for many languages.

Phrase-based machine translation (PBMT) is a form of SMT where the translation model aims to translate longer sequences of words ("phrases") in one go, solving part of the word ordering problem along the way that would be left to the decoder and the target language model in a word-based SMT system (Koehn et al., 2003). One advantage of PBMT is that it is adaptable to any language pair for which there is a parallel corpus available. The PBMT model makes use of a translation model, derived from the parallel corpus, and a language model, derived from a monolingual corpus in the target language. The language model is typically an n-gram model with smoothing. For any given input sentence, a search is carried out producing an n-best list of candidate translations, ranked by the decoder score, a complex scoring function including likelihood scores from the translation model and the target language model. In principle, all of this should be transportable to a data-driven machine translation account of paraphrasing. For this to work, however, a preferably large collection of data is required, which in this case would be pairs of sentences that paraphrase each other.

2.1.2 *Parallel corpora for paraphrasing*

Two recently published surveys on paraphrasing address the need for paraphrase corpora to further develop research into paraphrasing (Madnani and Dorr, 2010; Androutsopoulos and Malakasiotis, 2010). Androutsopoulos and Malakasiotis observe that not many such parallel corpora currently exist, and that the ones that do exist are not even close to the size of corpora generally used to train statistical machine translation systems (Androutsopoulos and Malakasiotis, 2010). Barzilay and McKeown suggest building parallel paraphrase corpora by using multiple human translations of literary works originally written in a different language (Barzilay and McKeown, 2001). The fact that different translators may use different wordings can be exploited to find paraphrase pairs within a language. In general, for the machine translation approach to paraphrasing to work, first the texts need to be aligned at the sentence level to obtain sentence pairs that can be used in a parallel monolingual corpus, where each sentence in translation T_1 is ideally seman-

tically equivalent to each sentence in translation T_2 in language L. Pang et al. use a similar approach to obtain paraphrases from the Multiple Translation Chinese corpus which contains eleven English translations of Chinese news articles on the sentence level (Pang et al., 2003).

Shinyama et al. use named entity recognition to extract paraphrases from various news articles describing the same event. The Microsoft Research Paraphrase Corpus (MSR) (Quirk et al., 2004; Dolan et al., 2004; Nelken and Shieber, 2006) is a paraphrase corpus constructed in an unsupervised manner. The MSR contains 5,801 pairs of sentences that were extracted from news sources on the Web, along with human annotations indicating whether each pair captures a paraphrase/semantic equivalence relationship. Of these sentences, the judges agreed that 67% were indeed paraphrases. Cohn et al. developed a monolingual parallel corpus consisting of 900 sentence pairs annotated with alignments at the word and phrase level, which also contained sentences from the MSR (Cohn et al., 2008).

In the field of textual entailment recognition efforts have also been made to construct textual entailment corpora. Burger and Ferro generated a corpus of textual entailment pairs by pairing sentences from the lead paragraph of news articles with the headlines of the articles. They were able to collect 100,000 entailment pairs in this manner.

While these corpora are valuable, for a statistical paraphrasing approach to work they are generally several orders too small. Preferably, such systems are trained on hundreds of thousands to millions of parallel sentences, where the available paraphrase corpora contain several thousand sentences at best. One solution to this problem is to leverage the abundance of bilingual parallel corpora to find paraphrases. Bannard and Burch use a bilingual corpus and a pivot language to construct a monolingual phrase-table (Bannard and Burch, 2005). They do this by aligning phrases across the two languages and then harvesting all phrases aligned to one phrase in the pivot language as paraphrases. In contrast to using a pivot language, we demonstrate that it is possible to construct a sufficiently large parallel corpus without relying on a second language, but by harvesting different headlines for the same event. This has several advantages. One reason to use headlines is that these are abundant on the Web in many languages, and every day new ones appear describing real world events. The real world knowledge implicitly present in the system stays up to date this way: it will know that in this time frame (early 2013) *"Barack Obama"* can be paraphrased as *"The President of the United States"*. A more crucial reason is that there is much paraphrastic variety in headlines. Different journalists and news editors will try to come up with their own unique headlines that describe the same event. Another reason is that we have less of a problem dealing with sentence alignment

between two texts to construct the parallel corpus, because headlines can be clustered relatively accurately by news aggregators such as Google News. Finally, headlines tend to be shorter than regular sentences and therefore words and phrases in them are easier to align.

2.1.3 *Evaluation*

As Callison-Burch et al. argue, automatic evaluation of paraphrasing is problematic (Callison-Burch et al., 2008). The essence of paraphrasing is to be able to generate a sentence that paraphrases a source, but that is at the same time structurally different from that source. Automatic evaluation metrics in related fields such as standard multilingual machine translation (e.g. BLEU (Papineni et al., 2002)) operate on a notion of joint semantic and structural similarity, while paraphrasing aims to achieve semantic similarity, but also structural dissimilarity. As Madnani and Dorr rightfully observe, precision and recall are not suited when evaluating sentential paraphrase generation, because no exhaustive list of paraphrases can exist (Madnani and Dorr, 2010). Madnani and Dorr state that semantic similarity and paraphrase recognition metrics can be applied to generated sentential paraphrases. Yet, besides semantic similarity there are more criteria that are applicable to paraphrases: they should also be grammatical, and they should be structurally dissimilar to the source sentence. There have been efforts to develop automatic metrics for the evaluation of paraphrases, such as ParaMetric (Callison-Burch, 2008) and PEM (Liu et al., 2010). ParaMetric is used to measure performance in alignment between two given sentences, and is not suited to measure the performance of a sentential paraphrase generation method given unseen sentences. PEM (Paraphrase Evaluation Metric) seems a promising approach in that it addresses the three crucial parts in paraphrase evaluation, namely fluency, adequacy and to some extent structural dissimilarity (PEM measures lexical dissimilarity). PEM makes no use of reference paraphrases; rather, it makes use of bilingual parallel corpora through the pivot approach. This suggests it might be biased towards paraphrasing approaches that use statistical machine translation and in particular pivot approaches. Another approach is to look at dissimilarity to the source sentence in addition to similarity to a collection of reference paraphrases. This is the approach we take and which has also been investigated by (Chen and Dolan, 2011). Chen and Dolan propose a new metric called PINC, which can be seen as a complement to BLEU: it measures the n-gram overlap between output and source sentence. The higher the overlap, the lower the PINC score. The idea is that good paraphrases show a high amount of overlap with refer-

ence paraphrases, and low overlap with the source sentence. We evaluate the output of our system by comparing it to a word substitution baseline, which uses a semantic lexicon and a language model to perform edit operations to construct a paraphrasing sentence, and a randomly selected human authored paraphrasing headline. We do this for Dutch and English and let 76 participants rate the paraphrases. We also take into account automatic machine translation evaluation metrics to see whether these correlate with human judgements, and show the results at different edit distances.

We expect the paraphrases generated by humans to score highest, and we consider this a soft upper bound for a paraphrasing system. We also expect that the PBMT-R system will perform better than the baseline, because it can do more complex paraphrasing operations than mere word substitution. In addition, we expect to see an effect of coverage: there are more headlines available for English than for Dutch, and also the semantic lexicon that we use for English has broader coverage than the one we use for Dutch. Therefore, we expect to see higher performance for the English systems than for the Dutch. Furthermore, we believe the extent to which a system paraphrases is an important aspect of its functioning. We expect that as the output of a paraphrasing system is increasingly dissimilar from the input sentence, the quality of the paraphrase is increasingly at risk, as the probability of paraphrasing errors increases. We expect that this detrimental effect will be smaller in the PBMT-R system and more pronounced in the baseline, as relatively few phrasal substitutions may lead to better paraphrases than relatively many single-word substitutions. On the topic of automatic measures, we think that the standard MT evaluation metrics will show some correlation to human judgement, but only to a small extent, as sentential paraphrasing is not simply about achieving similarity to a reference, and variations are probably greater in paraphrasing than in translating.

One way to solve this problem is to use a single monolingual corpus. From such a corpus one can extract semantically similar words using Harris' hypothesis of distributional similarity. (Lin and Pantel, 2001a) adapt the distributional similarity approach to dependency trees to find inference rules, resulting in the DIRT corpus. A drawback of this approach is that while the templates that are extracted often do express a textual entailment relation, this is often not a semantic equivalence as is needed for a paraphrase approach. Bilingual parallel corpora have the advantage that the two sides of the corpus are semantically equivalent. A popular method is to leverage the great number of multilingual parallel corpora to extract paraphrases. (Bannard and Burch, 2005) use a bilingual corpus and a pivot language to construct a monolingual phrase-table. They do this by aligning phrases across

the two languages and then harvesting all phrases aligned to one phrase in the pivot language as paraphrases. A logical alternative is to construct a true monolingual parallel corpus. (Barzilay and McKeown, 2001) built a parallel monolingual corpus using different translations of literary works.

2.2 DATA COLLECTION

For the development of our data collection method we use headline data from the DAESO corpus[1], a parallel monolingual treebank for Dutch (Marsi and Krahmer, 2007). Part of the data in the DAESO corpus consists of headline clusters crawled from Google News in the period April–August 2006. Google News uses clustering algorithms that consider the full text of each news article, as well as other features such as temporal and category cues, to produce sets of topically related articles. The crawler stores the headline and the first 150 characters of each news article crawled from the Google News Website. Roughly 13,000 clusters were retrieved. Table 2.1 shows part of a cluster. It is clear that although clusters deal roughly with one subject, the headlines can represent quite a different perspective on the content of the article; certain headlines are paraphrases, others are clearly not. To obtain only paraphrase pairs, the clusters need to be more coherent. In the DAESO project 865 clusters were manually subdivided into sub-clusters of headlines that show clear semantic overlap. Sub-clustering is no trivial task. Consider, for instance, the sentences in the example containing 'Afghanistan' or 'Uruzgan'. They can be seen as related to each other, but then the reader must know that Uruzgan is a province in Afghanistan where the Dutch UN force was stationed. Also, there are numerous headlines that cannot be subclustered with other headlines, such as the first three headlines shown in the example.

These automatically annotated data form the basis for our first goal, the development of a method to extract paraphrase pairs from headline clusters. We divide the annotated 865 headline clusters in a development set of 40 clusters, while the remaining 825 are used as test data. The headlines are stemmed using the Porter stemmer for Dutch (Kraaij and Pohlmann, 1994). Instead of a word overlap measure as used by (Barzilay and Elhadad, 2003), we use a modified TF.IDF word score as suggested by (Nelken and Shieber, 2006). Each sentence is viewed as a document, and each original cluster as a collection of documents. For each stemmed word i in sentence j, $TF_{i,j}$ is

1 http://daeso.uvt.nl/

Kamp : Veiligheid grootste probleem in Uruzgan *(Kamp: Security biggest problem in Uruzgan)*
Met gevechtsheli op Afghaanse theevisite *(With attack helicopter on Afghan tea-visit)*
Bevel overgedragen aan Nederlandse commandant *(Command transferred to Dutch commander)*
Nederlandse missie Uruzgan officieel begonnen *(Dutch mission Uruzgan officially started)* Nederlandse opbouwmissie in Afghanistan begint *(Dutch construction mission in Afghanistan begins)* Missie Uruzgan begonnen *(Mission Uruzgan has begun)* Eerste dag missie Uruzgan *(First day mission Uruzgan)*
Soldaten opbouwmissie Uruzgan keren terug *(Soldiers construction mission Uruzgan return)* Eerste militairen komen terug uit Afghanistan *(First servicemen come back from Afghanistan)* Eerste groep militairen Afghanistan keert terug *(First group of servicemen return from Afghanistan)* Kwartiermakers keren terug uit Uruzgan *(Quartermasters return from Uruzgan)*
Opgelucht onthaal van militairen uit Uruzgan *(Relieved welcome of servicemen from Uruzgan)* Opgelucht onthaal van Uruzgan-gangers *(Relieved welcome of Uruzgan-goers)*

Table 2.1: Part of a sample headline cluster crawled in August 2006 with English glosses. Each box represents a collection of headlines that can be considered paraphrases within the cluster.

a binary variable indicating if the word occurs in the sentence or not. The TF.IDF score can then be defined as follows:

$$\text{TF.IDF}_i = \text{TF}_{i,j} \cdot \log \frac{|D|}{|\{d_j : t_i \in d_j\}|}$$

$|D|$ is the total number of sentences in the cluster and $|\{d_j : t_i \in d_j\}|$ is the number of sentences that contain the term t_i. These scores are used in a vector space representation. The similarity between headlines can be calculated by using a cosine similarity function on the headline vectors.

type	precision	recall
k-means clustering	0.66	0.44
pairwise similarity	0.76	0.41

Table 2.2: Precision and recall for both alignment methods

2.2.1 *Clustering*

The first approach we investigate to align paraphrasing headlines is clustering. The original Google News headline clusters that we crawled are re-clustered into finer grained sub-clusters. We use the k-means implementation in the CLUTO[2] software package. The k-means algorithm assigns k centers to represent the clustering of n points ($k < n$) in a vector space. The total intra-cluster variance is minimized by the function

$$V = \sum_{i=1}^{k} \sum_{x_j \in S_i} (x_j - \mu_i)^2$$

where μ_i is the centroid of all the points $x_j \in S_i$.

The PK1 cluster-stopping algorithm as proposed by (Pedersen and Kulkarni, 2006) is used to find the optimal k for each sub-cluster:

$$\text{PK1}(k) = \frac{\text{Cr}(k) - mean(\text{Cr}[1...\delta K])}{\text{std}(\text{Cr}[1...\delta K])}$$

Here, Cr is a criterion function and δK is the minumum difference between two consecutive criterion values to stop clustering. As soon as $\text{PK1}(k)$ exceeds a threshold, $k-1$ is selected as the optimum number of clusters.

2 http://glaros.dtc.umn.edu/gkhome/views/cluto/

To find the optimal threshold value for cluster stopping, optimization is performed on the development data. The optimization function we use is an F-score:

$$F_\beta = \frac{(1+\beta^2)\cdot(\text{precision}\cdot\text{recall})}{(\beta^2\cdot\text{precision}+\text{recall})}$$

We count the number of alignments between possible paraphrases. For instance, in a cluster of four sentences, 6 alignments can be made. In our case, precision is the number of alignments retrieved from the clusters which are relevant, divided by the total number of retrieved alignments. Recall is the number of relevant retrieved alignments divided by the total number of relevant alignments.

We use an F_β-score for evaluation of the alignments. We favor precision over recall, because we are interested in obtaining correct alignments rather than obtaining many alignments. However, we do not want to optimize for precision alone, because we still want to retrieve a fair number of paraphrases and not only the ones that are very similar. Thus, we set $beta = 0.25$, which reflects our preference for precision. Through optimizing F_β-score on our development set, we find an optimal threshold for the PK1 algorithm $th_{pk1} = 1$. For each original cluster, k-means clustering is then performed using the k found by the cluster stopping function. In each newly obtained cluster all headlines can subsequently be aligned pair-wise.

2.2.2 *Pairwise similarity*

Another approach for aligning paraphrasing headlines is to directly calculate similarities for each pair of headlines within a cluster. If the similarity exceeds a certain threshold, the pair is accepted as a paraphrase pair. If it is below the threshold, it is rejected. However, as (Barzilay and Elhadad, 2003) have pointed out, this type of sentence alignment is only effective to a certain extent. Beyond that point, context is needed. With this in mind, we adopt two thresholds and the cosine similarity function to calculate the similarity between two sentences:

$$\cos(\theta) = \frac{V1\cdot V2}{\|V1\|\|V2\|}$$

where $V1$ and $V2$ are the word vectors of the two sentences being compared and $\|V1\|$ and $\|V1\|$ are the magnitudes of the vectors. If the similarity is higher than the upper threshold, it is accepted. If it is lower than the lower threshold, it is rejected. In the remaining case of a similarity between the two thresholds, similarity is calculated over the contexts of the two headlines, namely the text snippet that was retrieved with the headline. If this similarity

exceeds the upper threshold, it is accepted. Threshold values as found by optimizing on the development data using again an $F_{0.25}$-score, are $Th_{lower} = 0.2$ and $Th_{upper} = 0.5$. An optional final step is to add transitive alignments. For instance, if headline A is paired with headline B, and headline B is aligned to headline C, headline A can be aligned to C as well. We do not add these alignments, because when one incorrect alignment is made, this process adds a large number of incorrect alignments, particularly in large clusters.

2.2.3 *Alignment evaluation*

The 825 clusters in the test set contain 1,751 sub-clusters in total. In these sub-clusters there are 6,685 clustered headlines. Another 3,123 headlines are not part of a cluster. Table 2.2 displays the precision and recall of paraphrase detection of our two approaches. We observe that pairwise calculation of similarity with the back-off strategy of using context performs better than k-means clustering when we aim for higher precision. The k-means clustering approach suffers from outlying headlines that cannot be clustered. If we artificially ignore those headlines, k-means clustering achieves a precision of 0.91 and a recall of 0.43. However, because we do not know beforehand which headlines can be clustered and which cannot, we must conclude that k-means clustering performs worse than pairwise similarity in this setting.

2.2.4 *Obtaining headline paraphrase pairs*

We choose the pairwise similarity approach to extract paraphrasing headline pairs from new expanded datasets consisting of roughly 51,000 English headline clusters and 31,000 Dutch headline clusters, crawled from Google News in 2006 and in 2010. This method produces a collection of 9.3 million pairwise alignments of 1.9 million unique headlines for English and 841,588 pairwise alignments of 394,056 unique headlines for Dutch[3]. Example alignments created with this approach are given in Figure 2.1. To our knowledge this new paraphrase source is several orders larger than existing paraphrase corpora.

3 The aligned headline collection can be found at `http://ilk.uvt.nl/~swubben/resources.html`

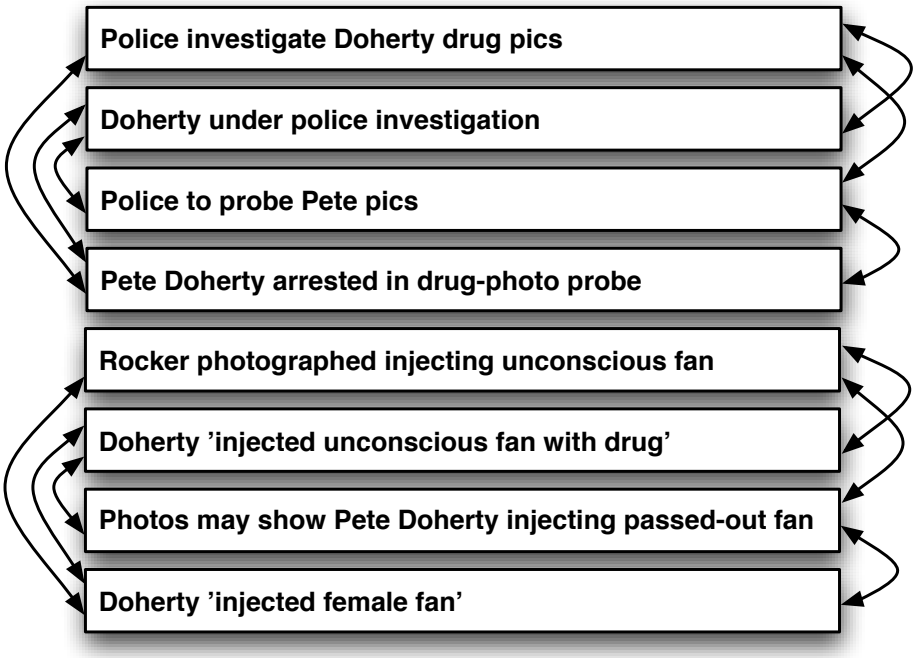


Figure 2.1: Part of a sample headline cluster, with aligned paraphrases

2.3 PARAPHRASE GENERATION

We use the collection of automatically obtained aligned headlines to train a paraphrase generation model using a phrase-based machine translation (PBMT) framework, extended with a post-hoc re-ranking model based on dissimilarity, resulting in our model PBMT-R. We compare this approach to a word substitution baseline. The generated paraphrases along with their source headlines are presented to human judges, whose ratings are compared to a collection of automatic machine translation evaluation metrics.

2.3.1 *PBMT-R*

We use the Moses software to train a PBMT model (Koehn et al., 2007). In general, a statistical machine translation model normally finds a best translation $\tilde{E}$ of a text in language F for a text in language E by combining a translation model $P(F|E)$ with a language model $P(E)$:

$$\tilde{E} = \arg\max_{E \in E^*} P(F|E)P(E)$$

In addition, in phrase-based machine translation the sentence F is segmented into a sequence of I phrases during decoding. Each source phrase can then be translated into a phrase in the target language to form sentence E. These phrases may be reordered. Phrase-based machine translation is described in more detail in Chapter 1.

The GIZA++ statistical alignment package is used to perform the word alignments, which are later combined into phrase alignments in the Moses pipeline (Och and Ney, 2003) to build the paraphrase model. GIZA++ implements IBM Models 1 to 5 and an HMM word alignment model to find statistically motivated alignments between words. We first tokenize our data before training a re-caser. We then lowercase all data and use all unique headlines in the training data to train an n-gram language model with the SRILM toolkit (Stolcke, 2002). Then we invoke the GIZA++ aligner using the training paraphrase pairs. We run GIZA++ with standard settings and we perform no optimization. Finally, we use the Moses decoder to generate paraphrases for our test data.

To expand the functionality of Moses in the intended direction we perform post-hoc re-ranking on the output based on dissimilarity to the input. We do this to select output that is as different as possible from the source sentence, so that ideally multiple phrases are paraphrased; at the same time, we base our re-ranking on a top-n of output candidates according to Moses, with a small n, to ensure that the quality of the output in terms of fluency and adequacy is also controlled for. Setting $n = 10$, for each source sentence we re-rank the ten best sentences as scored by the decoder according to the Levenshtein Distance (or edit distance) measure (Levenshtein, 1966) at the word level between the input and output sentence, counting the minimum number of edits needed to transform the source string into the target string, where the allowable edit operations are insertion, deletion, and substitution of a single word and casing is ignored. In case of a tie in Levenshtein Distance, we select the sequence with the better decoder score. When Moses is unable to generate ten different sentences, we select from the lower number of outputs. The resulting headlines are de-tokenized and re-cased using the previously trained re-caser.

2.3.2 *Word substitution baseline*

2.3.2.1 *English*

The PBMT-R results are compared with a word substitution baseline. For each noun, adjective and verb in the sentence this model takes that word and its part-of-speech tag and retrieves from the English WordNet (Fellbaum,

1998) all synonyms from all synsets the word occurs in. The English WordNet contains over 200K word-sense pairs. The word is then replaced by all of its synset words, and each replacement is scored by the trained SRILM language model also used in the PBMT-R system. The highest scoring alternative is kept. If no relevant alternative is found, the word is left unaltered. We use the Memory Based Tagger (Daelemans et al., 1996) trained on the Brown corpus to compute the part-of-speech tags. The WordNet::QueryData[4] Perl module is used to query WordNet.

2.3.2.2 *Dutch*

The word substitution baseline for Dutch works similarly to the English baseline and relies on the Cornetto database instead of WordNet. Cornetto is a lexical semantic database for Dutch, similar to WordNet. It includes 40K entries, covering the most generic and central part of the Dutch language (Vossen et al., 2008). As with the English system, all synonyms for a given word are extracted and the synonym which scores best in the sentence according to the language model is kept. The SRILM language model is trained on the Dutch headline paraphrase corpus.

2.4 EVALUATION

A human judgement study was set up to evaluate the generated paraphrases by both the baseline and the PBMT-R system, and to compare these with a human produced referent. The human judges rated both adequacy and fluency, and their judgements are compared to automatic evaluation measures in order to gain more insight into the automatic evaluation of paraphrasing.

2.4.1 *Method*

2.4.1.1 *Participants*

Participants were 76 students of Tilburg University, who participated for partial course credits. All were native speakers of Dutch, and all were proficient in English, having taken a course on Academic English at university level.

2.4.1.2 *Materials*

We randomly selected 1,000 headline clusters for Dutch and 1,000 headline clusters for English that appeared online in January 2011. Each cluster con-

4 `http://search.cpan.org/dist/WordNet-QueryData/QueryData.pm`

sisted of between 10 and 50 aligned paraphrasing headlines. We used these clusters as multiple references for our automatic evaluation measures to account for the diversity in real-world paraphrases, as the aligned paraphrased headlines in Figure 2.1 witness. For each participant we randomly selected 40 clusters, and from each cluster we randomly selected one headline as the source headline. Each headline was used as input for the word substitution baseline and the PBMT-R system, to generate two target paraphrases. In addition, we randomly selected one of the aligned headlines in a cluster to serve as the human produced upper bound to compare our systems with. For each source headline, we thus generated three target headlines (word substitution, PBMT-R, human-produced paraphrase). Each participant saw 40 different source headlines.

operation	sentences
single word replacement	50%
single word deletion or insertion	34%
word/phrase reordering	11%
phrase replacement	33%
sentence rewriting	2%

Table 2.3: Analysis of a sample of output from the English PBMT-R system indicating the number of sentences containing one or more of the specified edit operations.

2.4.1.3 *Procedure*

Participants were randomly assigned to the Dutch (N = 36) or English (N = 40) condition. In one version participants rated only Dutch target headlines, in the other they rated English ones. The instructions were otherwise identical for both versions. Participants were told that they participated in the evaluation of a system that could automatically generate headlines, and that they would see one source headline and three automatically generated paraphrases of that headline. They were not informed of the fact that we evaluated two different systems and always included one human reference headline. Following earlier evaluation studies (Doddington, 2002; Snover et al., 2009), we asked participants to evaluate both the fluency and adequacy of the target headlines on a five point Likert scale. Fluency was defined in the instructions as the extent to which a sentence reads well: is it good, clear Dutch or English? Adequacy was defined as the extent to which the

sentence is a good paraphrase of the example sentence. In the instruction this was illustrated with an example, where the source headline (for the English condition) was “Egyptian government orders Al Jazeera shutdown”, and examples were given of paraphrases that were adequate but not fluent (“Government Egypt want Al Jazeera stop"), and fluent but not adequate (“Egyptian government orders CNN shutdown"). If the instructions were clear, the experiment started. During the experiment, participants were in a sound-proof booth with a computer. Each source headline was presented on the computer screen, together with the three target headlines. The order of these targets on the screen was randomized, to prevent a bias towards one of the paraphrases. The experiment was individually performed, and self-paced; participants could take as much time as they required. On average the experiment lasted 33 minutes for English and 28 minutes for Dutch.

2.4.1.4 *Data analysis*

In total we collected 76 (participants) $\times$ 40 (clusters) $\times$ 3 (targets) = 9120 judgements. In practice, it turned out that the baseline system failed to generate a paraphrase in 12% of the cases for English and in 21% of the cases for Dutch. These could not be included in the analysis, so that the total number of collected judgements was lower. Since we are interested in the amount of edit operations a system performs and how these influence the evaluation, we computed the Levenshtein Distance (LD) from the source sentence of each target sentence at the word level ignoring casing. We created bins of LD 1, 2, 3, 4, and a collapsed bin of 5 or more to prevent data sparseness.

We performed two kinds of analyses. First we analyzed the human judgements in a by-item Multivariate Analysis of Variance (MANOVA) with Levenshtein Distance (levels: 1, 2, 3, 4, 5+), System (levels: word substitution, PBMT, human reference) and Language (levels: Dutch, English) as fixed factors and fluency and adequacy as dependent variables. Planned pairwise comparisons were made with the Bonferroni method.

Next, we evaluated the paraphrases using two automatic metrics, originating from the evaluation of machine translation: the BLEU (Papineni et al., 2002) and NIST (Doddington, 2002) metrics. BLEU measures n-gram overlap between strings, and is expressed as a score between 0 and 1, with higher scores representing more overlap. Different scores are calculated for n-grams of different size, up to n-grams of four. NIST is a BLEU variant giving more importance to less frequent (and hence more informative) n-grams. For each of the target paraphrases used in the evaluation experiment we compute BLEU and NIST scores, which we submitted to a MANOVA with the same design as used for the human judgements. We used the remaining headlines

for each cluster as the reference paraphrases for the automatic measures. In addition, we look at the correlations between the human judgements and the automatic metrics.

system	LD English	LD Dutch	fail rate English	fail rate Dutch
Word Sub	2.73	1.76	12%	21%
PBMT-R	2.57	2.88	0%	0%
Human	5.76	4.40	0%	0%

Table 2.4: Levenshtein distance and fail rate of output of the various systems

2.4.2 *Results*

Table 2.4 offers statistics showing the average LD of the target paraphrases in the cases where the system could find one, and the percentage of cases where the system was not able to generate a paraphrase of the source sentence. It can be observed that in general the PBMT-R system executes roughly equally many as the baseline for English, and more than the baseline for Dutch. Human produced paraphrases tend to differ more from the source. In addition, for 12 percent of the English sentences and 21 percent of the Dutch sentences the word substitution baseline could not provide a paraphrase. The PBMT-R system provided a paraphrase for every sentence.

2.4.2.1 *Human judgements*

Next, we analyzed the human judgements of fluency and adequacy of the target paraphrases. As expected, on both measures, the baseline word substitution system scored lowest (fluency: M_f = 2.86, adequacy: M_a = 2.61), and the randomly selected human reference scored highest (M_f = 4.18., M_a = 3.24) with the PBMT-R system sandwiched in between (M_f = 3.32, M_a = 2.86), showing a significant main effect for both fluency ($F(2, 7584) = 449.33, p < .001$) and adequacy ($F(2, 7584) = 95.48, p < .001$). All pairwise comparisons were statistically significant ($p < .001$). In addition, main effects were found for Language and Levenshtein Distance, but these are qualified by interactions with System. A significant interaction was found between Language and System, for both fluency ($F(2, 7584) = 147.93, p < .001$) and adequacy ($F(2, 7584) = 27.11, p < .001$). These interactions are illustrated in Figure 2.2, and suggest that the effect of language is larger for the Baseline

System	Headline
Source	U.S. Issues Call To Action To Support Breastfeeding
PBMT-R	US issues call to action for breastfeeding support
Word Sub	United States government Issues Phone call To Action To Support Breastfeeding
Source	Kuwait PM survives confidence vote
PBMT-R	Kuwaiti PM survives vote of confidence
Word Sub	Kuwait Autopsy survives assurance ballot
Source	John Edwards engaged to long-time mistress Rielle Hunter
PBMT-R	Edwards gets engaged to longtime girlfriend Rielle Hunter
Word Sub	Toilet Edwards engaged to long-time kept woman Rielle Huntsman
Source	Elizabeth Edwards' will makes no mention of her husband
PBMT-R	Elizabeth Edwards' to makes no mention of husband
Word Sub	Elizabeth Edwards' will makes no reference of her hubby
Source	Carlina White Update: Ann Pettway Admits She Stole Infant in 1987, Says FBI
PBMT-R	White Carlina Update: Ann pettway admits she stole baby in 1987 years, FBI says
Word Sub	Carlina White Update: Ann Pettway Admits She Stole Baby in 1987, Says Federal Bureau of Investigation
Source	Beckhams expecting their fourth child
PBMT-R	Beckham, Wife Expecting Third Child
Word Sub	Beckhams expecting their fourth kid

Table 2.5: Examples of generated English paraphrases where the PBMT-R system scores significantly better than the baseline (top) and where the baseline scores better (bottom)

than for the PBMT-R system, which might be due to the larger coverage of the English WordNet and the higher quality of the English language model. In addition, a significant interaction was found between Levenshtein Distance and System for fluency ($F(8, 7584) = 11.89, p < .001$), whereas the same interaction showed a trend towards significance for adequacy ($F(8, 7584) = 3.37, p = .071$). These effects are illustrated in Figure 2.3. First consider the results for fluency. It can be seen that fluency judgements of the human reference sentences do not vary with Levenshtein Distance, whereas the scores for the automatic systems show a steady decline as distance increases. Crucially, the performance of the PBMT-R system decreases less than the word substitution baseline beyond LD = 1. The picture for adequacy is slightly different: here all systems score lower as a function of LD, which is what one would expect given that the more distant a sentence is, the more likely it is that its content is also different. Crucially, however, while at LD = 1 the PBMT-R system scores roughly comparable to the baseline system, the two diverge more starting from LD = 2, and the PBMT-R system scores closer to the human reference than to the Baseline at LD = 5+.

2.4.2.2 *Automatic measures*

The results of the automatic evaluation metrics were analyzed next. We found that the baseline word substitution system attains the lowest scores (BLEU = 0.11, NIST = 7.00), and the randomly selected human reference scored highest (BLEU = 0.28, NIST = 8.19). We see that the PBMT-R system again scores between those two (BLEU = 0.18, NIST = 8.11), showing a significant effect for both BLEU ($F(2, 7584) = 200.91, p < .001$) and NIST ($F(2, 7584) = 105.54, p < .001$). In addition, main effects of Language and System are found, but these are again qualified by interactions.

Significant interactions between Levenshtein Distance and System were found for both BLEU ($F(8, 7584) = 5.790, p < .001$) and NIST ($F(8, 7584) = 14.070, p < .001$). These interactions can be explained by looking at Figure 2.4: at LD = 1, the word substitution baseline and the PBMT system score roughly comparable and substantially lower than the human referent. However, when considering larger distances, the scores show a decreasing trend, but the scores for the PBMT-R system drop less than those of the word substitution baseline. At LD = 5 the PBMT-R system scores very comparable to the human baseline; this pattern is especially pronounced for the NIST scores. In addition, significant interactions were found of Language and System, for both BLEU ($F(4, 7584) = 3.781, p < .01$) and NIST ($F(4, 7584) = 4.329, p < .01$), illustrated in Figure 2.5 respectively. This figure shows that, even though the PBMT-R system always scores higher than

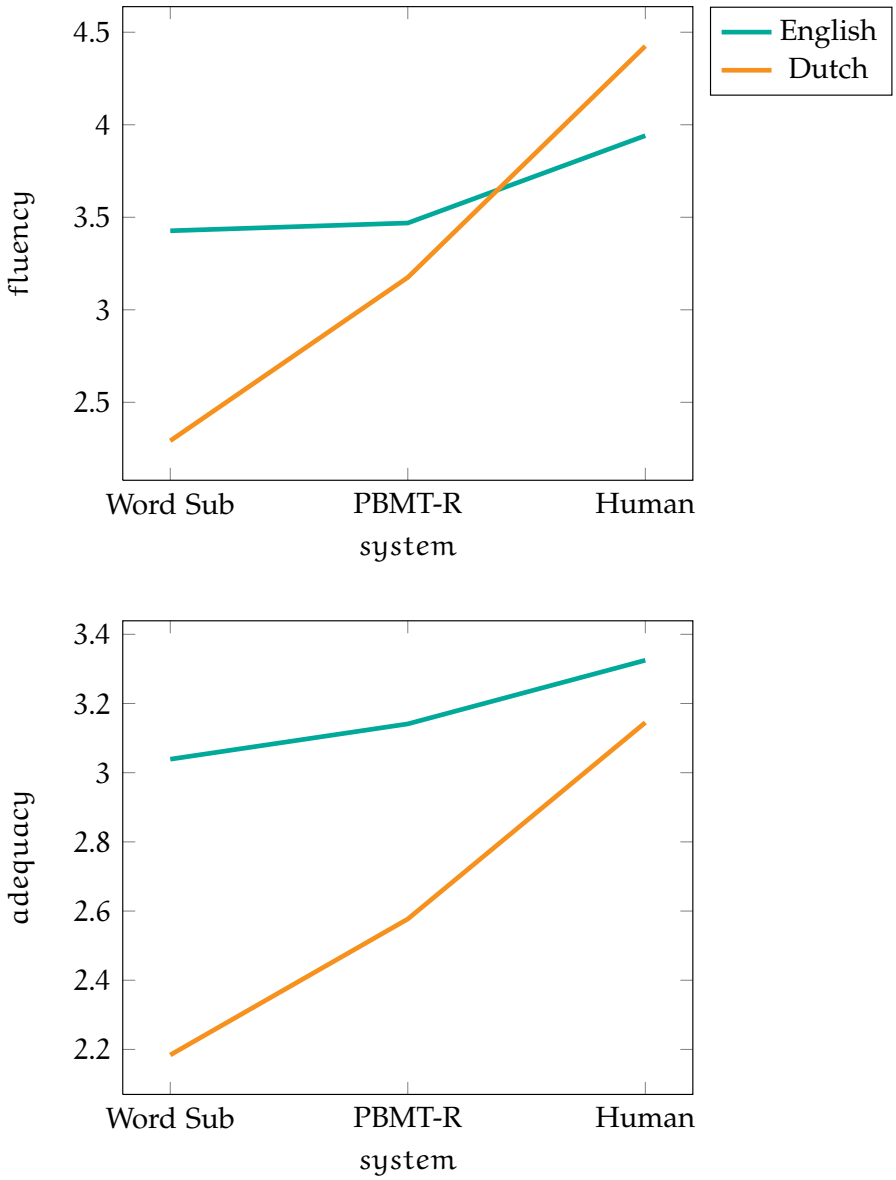


Figure 2.2: Fluency scores (top) and adequacy scores (bottom) per language as a function of system

the word substitution system, the difference is more pronounced for English than for Dutch.

In general, it is fair to say that the results of the automatic evaluation mirror those of the human judgements. This is confirmed by a correlation analysis. We found a strong correlation between BLEU and NIST, as expected, but, more interestingly, we also found that both correlate significantly and

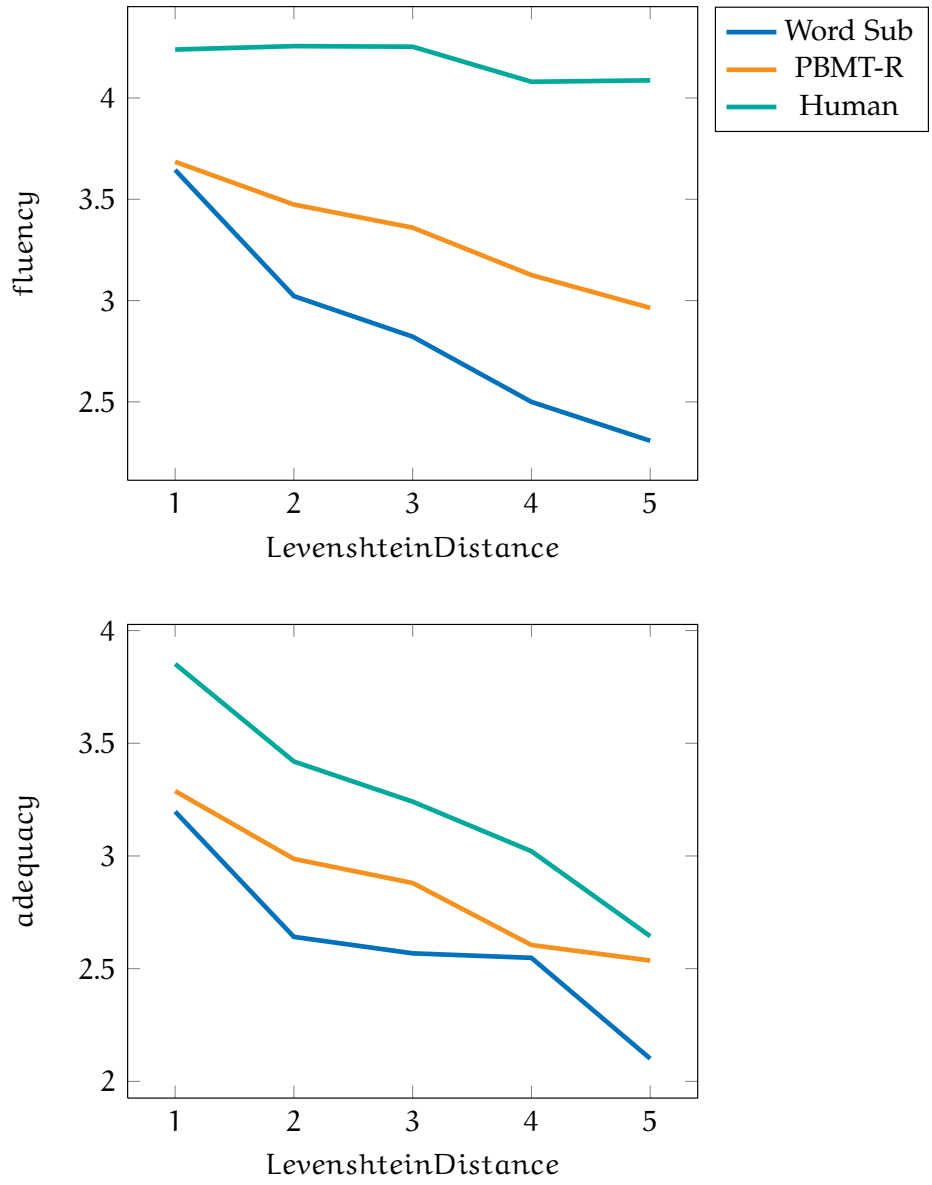


Figure 2.3: Fluency scores (top) and adequacy scores (bottom) per system as a function of Levenshtein Distance

positive with fluency ($r = .10$ for BLEU, and $r = .06$ for NIST, both $p < .001$) and adequacy ($r = .12$ for BLEU, and $r = .13$ for NIST, both $p < .001$).

Table 2.3 lists a breakdown of the paraphrasing operations the PBMT-R approach has performed. The number indicates the percentage of generated headlines out of a sample of 160 English generated headlines that contain one of the specified edit operations. Phrase replacements should be interpreted as a replacement involving multi-word phrases. Sentence rewriting

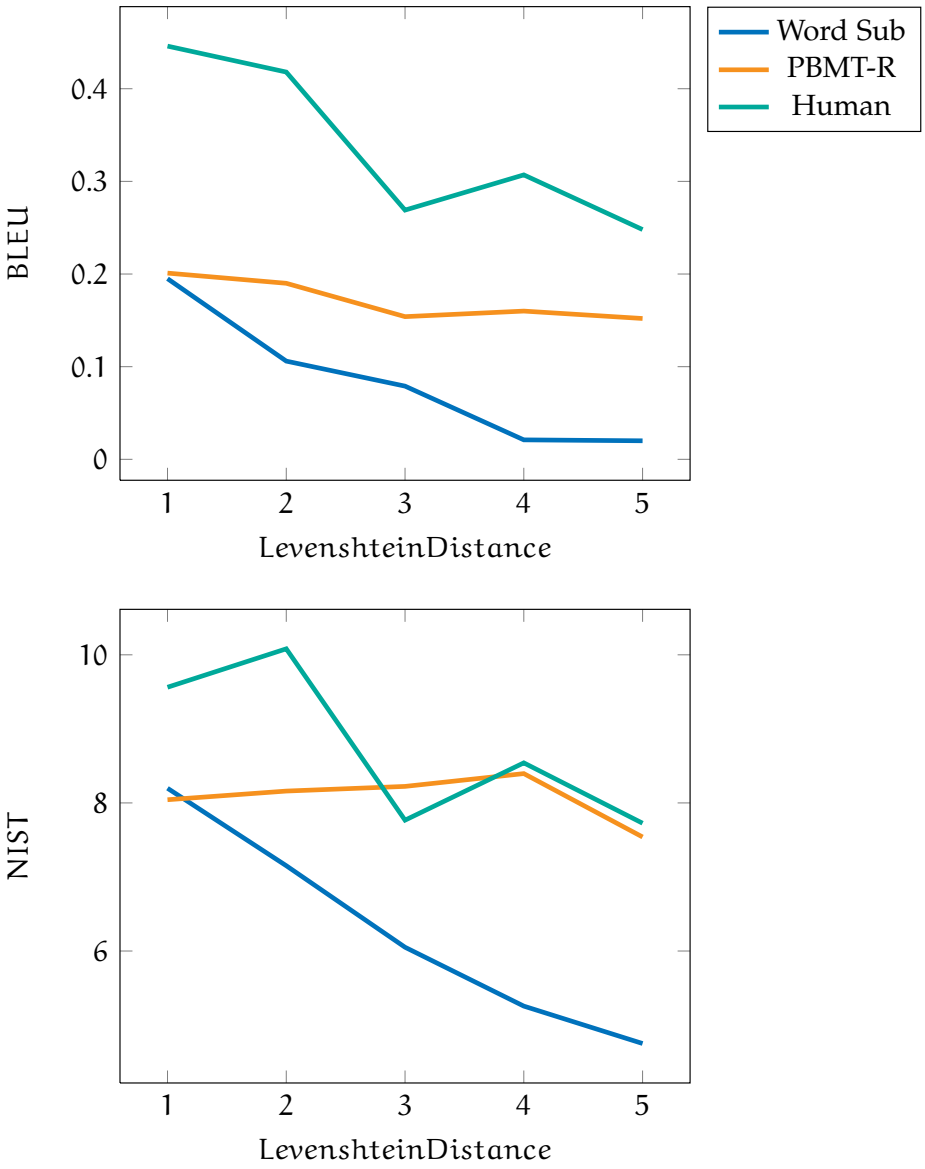


Figure 2.4: BLEU scores (top) and NIST scores (bottom) per system as a function of Levenshtein Distance

means that the sentence is fundamentally changed in its entirety, for instance changing from passive to active and vice versa. We observe that even though the PBMT-R system is capable of manipulating multi-word phrases, the most frequent change is still single word replacement, and a majority of changes involve single word edits (replacements, insertions, or deletions). Yet, a substantial number of changes made by the PBMT-R system involve more complex phrasal manipulations and re-orderings. Examples of generated English

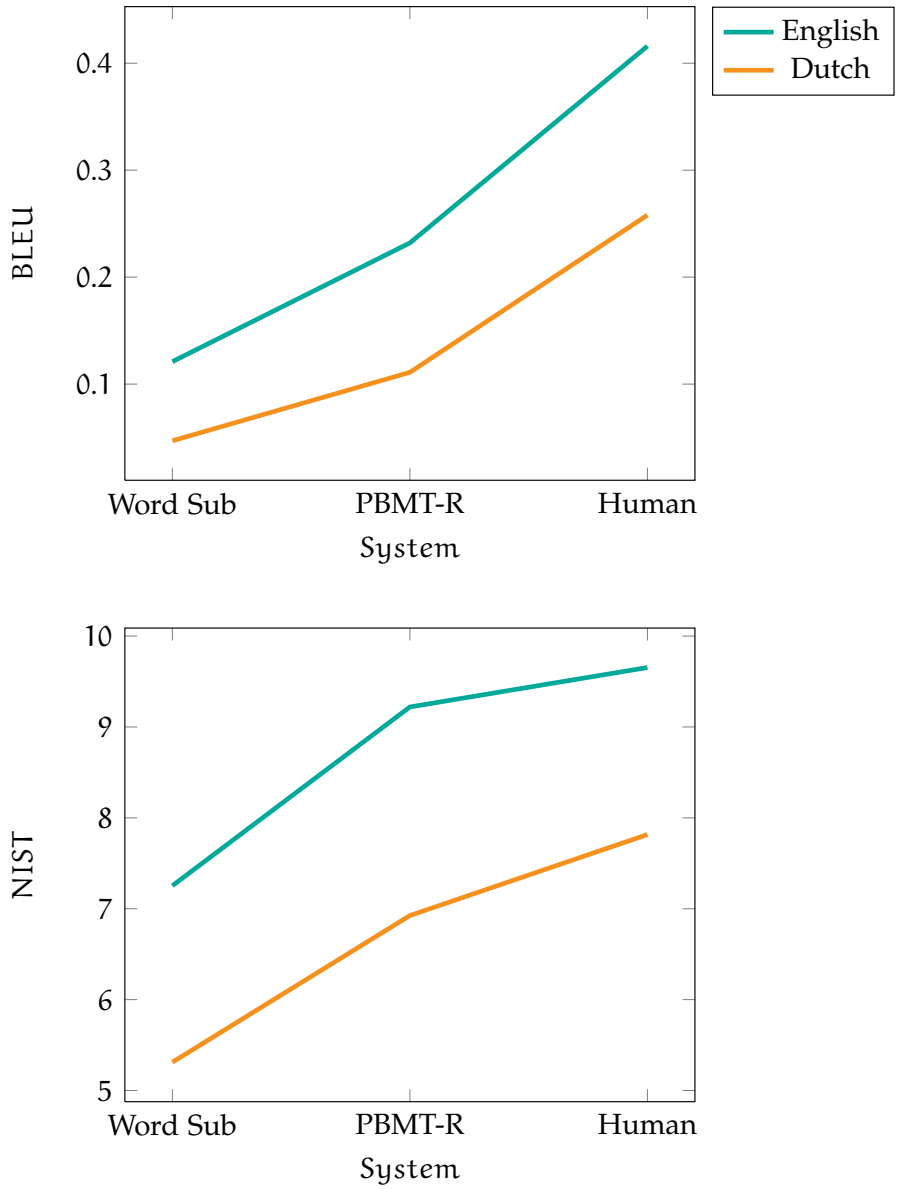


Figure 2.5: BLEU scores (top) and NIST scores (bottom) per language as a function of system

headlines are displayed in Table 2.5. The top three examples indicate headlines where the PBMT-R output received high scores and the baseline output received low scores, while the bottom three examples show headlines where the baseline found a better paraphrase than the PBMT-R system. In general the PBMT-R scores well when it finds correct phrase replacements and re-orderings, and the baseline scores badly when it substitutes words with irrelevant synonyms, or when it tries to substitute proper nouns that also

exist as common nouns ("John" and "Hunter" in the third example). Some interesting errors can be observed in the bottom three examples. Here, the PBMT-R system makes different kinds of errors: in the fourth example the output is ungrammatical ("to makes"), in the fifth example we observe an erroneous re-ordering ("White Carlina") and insertion "1987 years", and in the last example the PBMT-R system makes an error in content, replacing "fourth" with "third".

2.5 CONCLUSION AND DISCUSSION

In this chapter we have presented a method to build a corpus of aligned sentential paraphrases. We have created this corpus by crawling a news aggregator site (in our case Google News) and aligning these headlines based on overlap. We used a standard PBMT framework with a dissimilarity component to generate the output paraphrases for two languages, English and Dutch, and compared this approach to a word substitution baseline.

In general, we found that the PBMT-R system outperforms the word substitution system on all dimensions of evaluation: it always succeeds in generating a paraphrase, while the baseline system fails to do so on 12% (English) to 21% (Dutch) of the source sentences. If we concentrate on the cases where the baseline system succeeds in generating a paraphrase, we find that the PBMT-R paraphrases are on average more dissimilar to the source sentences, as shown by their higher average Levenshtein distance. The human evaluators rated the output of the PBMT-R system higher than that of the baseline system, both in terms of adequacy and fluency. The human judgements show that while the performance of the baseline system drops substantially with higher Levenshtein distances, the PBMT-R system shows a less steep decline on both dimensions of evaluation. The automatic evaluation metrics (BLEU and NIST) reveal a similar pattern.

Even though the general picture is remarkably consistent, a number of language effects are found. In particular, we can see that the word substitution baseline for English performs better than for Dutch (both according to the human judges, and in terms of the automatic metrics). This is not unexpected since the English semantic lexical resource has a higher coverage than its Dutch counterpart (which is also reflected by the higher percentage of cases in which the baseline system fails to generate a paraphrase), and that the language model for English may be better than that for Dutch as it is trained on more headline data. We would like to stress that the word substitution baseline is an informed baseline, which is nevertheless improved

upon by the PBMT-R system. At the same time, the scores for human reference paraphrases reveal that there is still substantial room for improvement.

The use of a PBMT framework combined with the exploitation of crawled corpora of aligned headlines is a feasible strategy; human judges preferred the output of our PBMT-R system over the output of the word substitution system. However, it should be noted that the fluency of the PBMT-R system output is still very much below the fluency of human produced headlines. We have also addressed the problem of automatic paraphrase evaluation. We measured BLEU and NIST scores, and observed that these automatic scores correlate with human judgements to some degree. Overall they show the same picture: the selected human paraphrase scores best, followed by the PBMT-R system and the word substitution baseline comes in last. Because standard MT metrics such as BLEU and NIST do not take into account the notion of dissimilarity, these scores tend be high when few edits are made and drop as the paraphrases deviate more from the source sentence. When edit distance is considered, the decline of the scores of different systems can be compared.

We feel that our approach of using a corpus of crawled and aligned headlines together with an off-the-shelf PBMT package, modified to re-rank on dissimilarity (PBMT-R), is an important contribution in paraphrase research, as it allows the research to extend beyond English. We note that automatic evaluation of sentential paraphrase generation is a very important problem. Not only should a generated paraphrase be judged by its semantic similarity to the input sentence (adequacy), but also on its fluency, and it should obey the constraint that it be structurally different from the input sentence. The interactions between adequacy, fluency, and structural dissimilarity deserve more research. It might be feasible to modify MT evaluation metrics to automatically evaluate generated paraphrastic sentences.

It is worth noting that headlines sometimes use a particular kind of language. This "headlinese" is often shorter than regular language, apparent in its syntax where for instance articles or verbs can be omitted. In certain contexts headlines can contain puns or jokes which might also translate badly in other context. Despite the current bias to the headline genre, we believe the results show that an advantage of our approach is that it is able to paraphrase entire sentences: it paraphrases those parts of sentences that it can paraphrase, and leaves other parts intact, such as parts it has not encountered before. This is different from bilingual translation, where all material needs to be translated, even when some of the input is unknown. In this sense, paraphrasing as monolingual machine translation is an easier task than multilingual translation, and our results indicate that our system indeed performs reasonably well. However, it can also be argued that para-

phrasing has a harder aspect: in multilingual translation, all we need to do is to find a valid translation. In paraphrasing, we need to find a valid translation and this translation needs to be sufficiently different from the source sentence.

Our system, trained on the corpus of crawled and aligned headlines, may be usable in other domains and genres as well; it may be possible to train a language model on text from the new domain, and use the translation model acquired from the headlines to generate paraphrases for the new domain. We are also interested in capturing other monolingual text-to-text data, such as simplification or compression data, but acquiring monolingual parallel corpora for different domains is no trivial task.

3

SENTENCE SIMPLIFICATION

In this chapter we describe a method for simplifying sentences using Phrase-based machine translation (PBMT), augmented with a re-ranking heuristic based on dissimilarity (PBMT-R), and trained on a monolingual parallel corpus. We compare our system to a word-substitution baseline and two state-of-the-art systems, all trained and tested on paired sentences from the English part of Wikipedia and Simple Wikipedia. Human test subjects judge the output of the different systems. Analysis of the judgements shows that by relatively careful phrase-based paraphrasing our model achieves similar simplification results to state-of-the-art systems, while generating better formed output. We also argue that text readability metrics such as the Flesch-Kincaid grade level should be used with caution when evaluating the output of simplification systems.

THIS CHAPTER IS BASED ON: Wubben, S., van den Bosch, A.P.J., & Krahmer, E.J. (2012). *Sentence simplification by monolingual machine translation* , Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (pp. 1015-1024). Jeju, Republic of Korea: Association for Computational Linguistics

3.1 INTRODUCTION

Sentence simplification can be defined as the process of producing a simplified version of a sentence by changing some of the lexical material and optionally the grammatical structure of that sentence, while still preserving the semantic content of the original sentence, in order to ease its understanding. Language learners (Siddharthan, 2002), people with reading disabilities (Inui et al., 2003) such as aphasia (Carroll et al., 1999), and low-literacy readers (Watanabe et al., 2009) could especially benefit from this application. It can serve to generate output in a specific limited format, such as subtitles (Daelemans et al., 2004). Sentence simplification can also serve to preprocess the input of other tasks, such as summarization (Knight and Marcu, 2000), parsing, machine translation (Chandrasekar et al., 1996), semantic role labeling (Vickrey and Koller, 2008) or sentence fusion (Filippova and Strube, 2008).

The main goal of simplification is to make texts more accessible, for instance for low literacy readers. Some of the factors that are known to help this process are the vocabulary used, the length of the sentences, the syntactic structures present in the text, and the usage of discourse markers. One effort to create a simple version of English at the vocabulary level was the creation of Basic English by Charles Kay Ogden (Ogden and Graham, 1935). Basic English is a controlled language with a basic vocabulary consisting of 850 words. According to Ogden, 90 percent of all dictionary entries can be paraphrased using these 850 words. An example of a resource that is written using mainly Basic English is the English Simple Wikipedia. Articles in English Simple Wikipedia are similar to articles found in the traditional English Wikipedia, but written using a limited vocabulary (using Basic English where possible). Examples of sentences from normal English Wikipedia and Simple English Wikipedia are displayed in Table 3.1. Generally, the structure of the sentences in English Simple Wikipedia is less complicated and the sentences are somewhat shorter than those found in English Wikipedia. We offer more detailed statistics below.

3.1.1 *Related work*

Most earlier work on sentence simplification adopted rule-based approaches. A frequently applied type of rule, aimed to reduce overall sentence length, splits long sentences on the basis of syntactic information (Chandrasekar and Srinivas, 1997; Carroll et al., 1998; Canning et al., 2000; Vickrey and Koller, 2008). Siddharthan (2011) investigate regeneration from typed de-

pendencies for sentence simplification. There has also been work on lexical substitution for simplification, where the aim is to substitute difficult words with simpler synonyms, derived from WordNet or dictionaries (Inui et al., 2003). Another venue of research on text simplification is question generation (Heilman and Smith, 2010).

Zhu et al. (2010) examine the use of paired documents in English Wikipedia and Simple Wikipedia for a data-driven approach to the sentence simplification task. They propose a probabilistic, syntax-based machine translation approach to the problem and compare against a baseline of no simplification and a phrase-based machine translation approach. In a similar vein, Coster and Kauchak (2011) use a parallel corpus of paired documents from Simple Wikipedia and Wikipedia to train a phrase-based machine translation model coupled with a deletion model. Another useful resource is the edit history of Simple Wikipedia, from which simplifications can be learned (Yatskar et al., 2010). Woodsend and Lapata (2011) investigate the use of Simple Wikipedia edit histories and an aligned Wikipedia–Simple Wikipedia corpus to induce a model based on quasi-synchronous grammar. They select the most appropriate simplification by using integer linear programming.

English Wikipedia:	Blade Runner initially polarized critics: some were displeased with the pacing, while others enjoyed its thematic complexity. The film performed poorly in North American theaters but has since become a cult film. The film has been hailed for its production design, depicting a "retrofitted" future, and remains a leading example of the neo-noir genre.
Simple Wikipedia:	Some movie critics did not like Blade Runner because they thought it was slow, but others liked its many ideas. The movie did not sell many tickets in North American movie theaters but was more popular in other countries. Even though it did not make much money, it was liked very much by teachers and science fiction fans. Blade Runner looked good and made the future look very dark and old.

Table 3.1: Example sentences from articles from normal English Wikipedia and Simple English Wikipedia.

We follow Zhu et al. (2010) and Coster and Kauchak (2011) in proposing that sentence simplification can be approached as a monolingual machine

translation task, where the source and target languages are the same and where the output should be simpler in form than the input but similar in meaning. Generated output should be well formed. This means the output sentences should be fluent and still similar in meaning as the input sentences. We differ from the approach of Zhu et al. (2010) in the sense that we do not take syntactic information into account; we rely on PBMT to do its work and implicitly learn simplifying paraphrasings of phrases. Our approach differs from Coster and Kauchak (2011) in the sense that instead of focusing on deletion in the PBMT decoding stage, we focus on dissimilarity, as simplification does not necessarily imply shortening (Woodsend and Lapata, 2011), or as the Simple Wikipedia guidelines state, "simpler does not mean short"[1]. Table 3.2 shows the average sentence length and the average word length for Wikipedia and Simple Wikipedia sentences in the PWKP dataset used in this study (Zhu et al., 2010). These numbers suggest that, although the selection criteria for sentences to be included in this dataset are biased (see Section 2.2), Simple Wikipedia sentences are about 17% shorter, while the average word length is virtually equal.

source	sent. length	token length
Simple Wikipedia	20.87	4.89
Wikipedia	25.01	5.06

Table 3.2: Average sentence and token length statistics for the PWKP dataset (Zhu et al., 2010).

Statistical machine translation (SMT) has already been successfully applied to the related task of paraphrasing (Quirk et al., 2004; Bannard and Burch, 2005; Madnani et al., 2007; Callison-Burch, 2008; Zhao et al., 2009; Wubben et al., 2010). SMT typically makes use of large parallel corpora to train a model on. These corpora need to be aligned at the sentence level. Large parallel corpora, such as the multilingual proceedings of the European Parliament (Europarl), are readily available for many languages. Phrase-Based Machine Translation (PBMT) is a form of SMT where the translation model aims to translate longer sequences of words ("phrases") in one go, solving part of the word ordering problem along the way that would be left to the target language model in a word-based SMT system. PMBT operates purely on statistics and no linguistic knowledge is involved in the process: the phrases that are aligned are motivated statistically, rather than linguistically. This makes PBMT adaptable to any language pair for which there

1 http://simple.wikipedia.org/wiki/Main_Page/Introduction

is a parallel corpus available. The PBMT model makes use of a translation model, derived from the parallel corpus, and a language model, derived from a monolingual corpus in the target language. The language model is typically an n-gram model with smoothing. For any given input sentence, a search is carried out producing an n-best list of candidate translations, ranked by the decoder score, a complex scoring function including likelihood scores from the translation model, and the target language model. In principle, all of this should be transportable to a data-driven machine translation account of sentence simplification, provided that a parallel corpus is available that pairs text to simplified versions of that text.

3.1.2 *This study*

In this chapter we investigate the use of phrase-based machine translation modified with a dissimilarity component for the task of sentence simplification. While Zhu et al. (2010) have demonstrated that their approach outperforms a PBMT approach in terms of Flesch Reading Ease test scores, we are not aware of any studies that evaluate PBMT for sentence simplification with human judgements. In this study we evaluate the output of Zhu et al. (2010) (henceforth referred to as 'Zhu'), Woodsend and Lapata (2011) (henceforth referred to as 'RevILP'), our PBMT based system with dissimilarity-based reranking (henceforth referred to as 'PBMT-R'), a word-substitution baseline, and, as a gold standard, the original Simple Wikipedia sentences. We will first discuss the baseline, followed by the Zhu system, the RevILP system, and our PBMT-R system in Section 3.2. We then describe the experiment with human judges in Section 3.3, and its results in Section 3.4. We close this chapter by critically discussing our results in Section 4.5.

3.2 SENTENCE SIMPLIFICATION MODELS

3.2.1 *Word-substitution baseline*

The word substitution baseline replaces words in the source sentence with (near-)synonyms that are more likely according to a language model. For each noun, adjective and verb in the sentence this model takes that word and its part-of-speech tag and retrieves from WordNet all synonyms from all synsets the word occurs in. The word is then replaced by all of its synset words, and each replacement is scored by a SRILM language model (Stolcke, 2002) with probabilities that are obtained from training on the Simple Wikipedia data. The alternative that has the highest probability according to

the language model is kept. If no relevant alternative is found, the word is left unchanged. We use the Memory-Based Tagger (Daelemans et al., 1996) trained on the Brown corpus to compute the part-of-speech tags. The WordNet::QueryData[2] Perl module is used to query WordNet (Fellbaum, 1998).

3.2.2 *Zhu et al.*

Zhu et al. (2010) learn a sentence simplification model which is able to perform four rewrite operations on the parse trees of the input sentences, namely substitution, reordering, splitting, and deletion. Their model is inspired by syntax-based SMT (Yamada and Knight, 2001) and consists of a language model, a translation model and a decoder. The four mentioned simplification operations together form the translation model. They use an expectation maximization (EM) algorithm to train the model iteratively. Their decoder greedily generates simplifications with the help of a language model. It is trained on a corpus containing aligned sentences from English Wikipedia and English Simple Wikipedia called PWKP. The PWKP dataset consists of 108,016 pairs of aligned lines from 65,133 Wikipedia and Simple Wikipedia articles. These articles were paired by following the "interlanguage link"[3]. TF*IDF at the sentence level was used to align the sentences in the different articles (Nelken and Shieber, 2006).

Zhu et al. (2010) evaluate their system using BLEU and NIST scores, as well as various readability scores that only take into account the output sentence, such as the Flesch Reading Ease test and n-gram language model perplexity. Although their system outperforms several baselines at the level of these readability metrics, they do not achieve better scores when evaluated with BLEU or NIST.

3.2.3 *RevILP*

Woodsend and Lapata (2011)'s model is based on quasi-synchronous grammar (Smith and Eisner, 2006). Quasi-synchronous grammar generates a loose alignment between parse trees. It operates on individual sentences annotated with syntactic information in the form of phrase structure trees, which Woodsend and Lapata obtain from parsing with the Stanford parser. Quasi-synchronous grammar is used to generate all possible rewrite operations. Each QG rule describes the transformations necessary to morph a source subtree into a target subtree. This allows child nodes to be deleted, re-

2 http://search.cpan.org/dist/WordNet-QueryData/QueryData.pm

3 http://en.wikipedia.org/wiki/Help:Interlanguage_links

ordered or flattened. Additionally, extra operations are allowed, such as the introduction of punctuation. Sentence simplification rules are then learned from the aligned sub-trees. Sentence splitting rules are learned from alignments where one source sentence is aligned to two target sentences. Their strategy is to identify a node in the source sentence that contributes to both target sentences. One is then dubbed the main sentence and the other the auxiliary sentence. The resulting QG rule is a tuple aligning the source node with a target node in the main sentence, and the entire structure of the auxiliary sentence.

After alignment, the next step is then to employ integer linear programming to generate and select the most appropriate simplification. This process starts at the root node of the parse tree, and applies QG rules to subtrees until the leaf nodes are reached. Woodsend and Lapata train two models: one model is trained on data containing alignments between Wikipedia and English Simple Wikipedia (AlignILP), and the other model is trained on data containing alignments between edits in the revision history of Simple Wikipedia (RevILP). RevILP performs best according to the human judgements conducted in their study. They show that it achieves better scores than Zhu et al. (2010)'s system and is not scored significantly differently from English Simple Wikipedia. In this study we compare our approach to their best performing model, the RevILP model.

3.2.4 *PBMT-R*

We use the Moses software to train a PBMT model (Koehn et al., 2007). The data we use is the PWKP dataset created by Zhu et al. (2010). In general, a statistical machine translation model finds a best translation $\tilde{E}$ of a sentence in one language F to a sentence in another language E by combining a translation model that finds the most likely translation $P(F|E)$ with a language model that outputs the most likely sentence $P(E)$:

$$\tilde{E} = \arg\max_{E \in E^*} P(F|E)P(E)$$

In phrase-based machine translation the sentence F is segmented into a sequence of I phrases during decoding. Each phrase is then translated into a phrase to form sentence E. Phrases may be reordered. This is described in more detail in Chapter 1.

The GIZA++ statistical alignment package is used to perform the word alignments, which are later combined into phrase alignments in the Moses pipeline (Och and Ney, 2003) to build the sentence simplification model. GIZA++ utilizes IBM Models 1 to 5 and an HMM word alignment model to

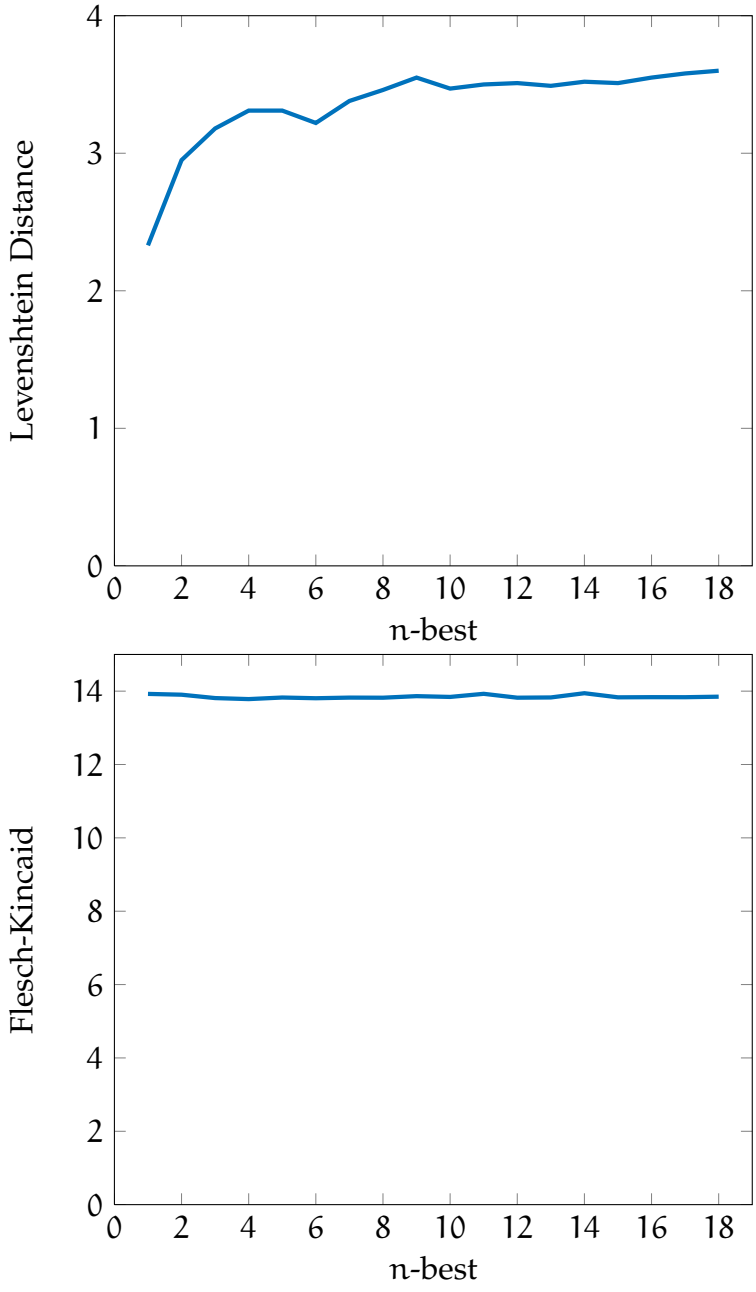


Figure 3.1: Levenshtein distance and Flesch-Kincaid score of output when varying the n of the n-best output of Moses.

find statistically motivated alignments between words. We first tokenize and lowercase all data and use all unique sentences from the Simple Wikipedia part of the PWKP training set to train an n-gram language model with the SRILM toolkit to learn the probabilities of different n-grams. Then we invoke

the GIZA++ aligner using the training simplification pairs. We run GIZA++ with standard settings and we perform no optimization. This results in a phrase table containing phrase pairs from Wikipedia and Simple Wikipedia and their conditional probabilities as assigned by Moses. Finally, we use the Moses decoder to generate simplifications for the sentences in the test set. For each sentence we let the system generate the ten best distinct solutions (or less, if fewer than ten solutions are generated) as ranked by Moses.

Arguably, dissimilarity is a key factor in simplification (and in paraphrasing in general). As output we would like to be able to select fluent sentences that adequately convey the meaning of the original input, yet that contain differences that operationalize the intended simplification. When training our PBMT system on the PWKP data we may assume that the system learns to simplify automatically, yet there is no aspect of the decoder function in Moses that is sensitive to the fact that it should try to be different from the input – Moses may well translate input to unchanged output, as much of our training data consists of partially equal input and output strings.

To expand the functionality of Moses in the intended direction we perform post-hoc re-ranking on the output based on dissimilarity to the input. We do this to select output that is as different as possible from the source sentence, so that it ideally contains multiple simplifications; at the same time, we base our re-ranking on a top-n of output candidates according to Moses, with a small n, to ensure that the quality of the output in terms of fluency and adequacy is also controlled for. Setting $n = 10$, for each source sentence we re-rank the ten best sentences as scored by the decoder according to the Levenshtein Distance (or edit distance) measure (Levenshtein, 1966) at the word level between the input and output sentence, counting the minimum number of edits needed to transform the source string into the target string, where the allowable edit operations are insertion, deletion, and substitution of a single word. In case of a tie in Levenshtein Distance, we select the sequence with the better decoder score. When Moses is unable to generate ten different sentences, we select from the lower number of outputs. Figure 3.1 displays Levenshtein Distance and Flesch-Kincaid grade level scores for different values of n. We use the Lingua::EN::Fathom module[4] to calculate Flesch-Kincaid grade level scores. The Flesch-Kincaid grade level score indicates the readability of a text and is defined as follows:

$$\mathrm{FK}_{gradelevel} = 206.835 - 1.015\left(\frac{\text{total words}}{\text{total sentences}}\right) - 84.6\left(\frac{\text{total syllables}}{\text{total words}}\right)$$

4 http://http://search.cpan.org/~kimryan/Lingua-EN-Fathom-1.15/lib/Lingua/EN/Fathom.pm

Figure 3.1 shows that the readability score stays more or less the same, indicating no relation between n and readability. The average edit distance starts out at just above 2 when selecting the 1-best output string, and increases roughly until $n = 10$.

3.2.5 *Descriptive statistics*

Table 3.3 displays the average edit distance and the percentage of cases in which no edits were performed for each of the systems and for Simple Wikipedia. We see that the Levenshtein distance between Wikipedia and Simple Wikipedia is the most substantial with an average of 12.3 edits. Given that the average number of tokens is about 25 for Wikipedia and 21 for Simple Wikipedia (cf. Table 3.2), these numbers indicate that the changes in Simple Wikipedia go substantially beyond the average four-word length difference. On average, eight more words are interchanged for other words. About half of the original tokens in the source sentence do not return in the output. Of the three simplification systems, the Zhu system (7.95) and the RevILP (7.18) attain similar edit distances, less substantial than the edits in Simple Wikipedia, but still considerable compared to the baseline word-substitution system (4.26) and PBMT-R (3.08). Our system is clearly conservative in its edits.

system	LD	perc. no edits
Simple Wikipedia	12.30	3
Word Sub	4.26	0
Zhu	7.95	2
RevILP	7.18	22
PBMT-R	3.08	5

Table 3.3: Levenshtein Distance and percentage of unaltered output sentences.

On the other hand, we observe some differences in the percentage of cases in which the systems decide to produce a sentence identical to the input. In 22 percent of the cases the RevILP system does not alter the sentence. The other systems make this decision about as often as the gold standard, Simple Wikipedia, where only 3% of sentences remain unchanged. The word-substitution baseline always manages to make at least one change.

3.3 EVALUATION

3.3.1 *Participants*

Participants were 46 students of Tilburg University, who participated for partial course credits. All were native speakers of Dutch, and all were proficient in English, having taken a course on Academic English at University level.

3.3.2 *Materials*

We use the test set used by Zhu et al. (2010) and Woodsend and Lapata (2011). This test set consists of 100 sentences from articles found in English Wikipedia, paired with sentences from corresponding articles in English Simple Wikipedia. We selected only those sentences where every system would perform minimally one edit, because we only want to compare the different systems when they actually generate altered, assumedly simplified output. From this subset we randomly pick 20 source sentences, resulting in 20 clusters of one source sentence and 5 simplified sentences, as generated by humans (Simple Wikipedia) and the four systems.

3.3.3 *Procedure*

The participants were told that they participated in the evaluation of a system that could simplify sentences, and that they would see one source sentence and five automatically simplified versions of that sentence. They were not informed of the fact that we evaluated in fact four different systems and the original Simple Wikipedia sentence. Following earlier evaluation studies (Doddington, 2002; Woodsend and Lapata, 2011), we asked participants to evaluate simplicity, fluency and adequacy of the target headlines on a five point Likert scale. Fluency was defined in the instructions as the extent to which a sentence is proper, grammatical English. Adequacy was defined as the extent to which the sentence has the same meaning as the source sentence. Simplicity was defined as the extent to which the sentence was simpler than the original and thus easier to understand. The order in which the clusters had to be judged was randomized and the order of the output of the various systems was randomized as well.

3.3.4 *Automatic measures*

The results of the automatic measures are displayed in Table 3.4. In terms of the Flesch-Kincaid grade level score, where lower scores are better, the Zhu system scores best, with 7.86 even lower than Simple Wikipedia (8.57). Increasingly poor Flesch-Kincaid scores are produced by RevILP (8.61) and PBMT-R (13.38), while the word substitution baseline scores worst (14.64). With regard to the BLEU score, where Simple Wikipedia is the reference, the PBMT-R system scores highest with 0.43, followed by the RevILP system (0.42) and the Zhu system (0.38). The word substitution baseline scores lowest with a BLEU score of 0.34.

system	Flesch-Kincaid	BLEU
Simple Wikipedia	8.57	1
Word Sub	14.64	0.34
Zhu	7.86	0.38
RevILP	8.61	0.42
PBMT-R	13.38	0.43

Table 3.4: Flesch-Kincaid grade level and BLEU scores

3.4 RESULTS

system	overall	fluency	adequacy	simplicity
Simple Wikipedia	3.46 (0.39)	3.84 (0.46)	2.91 (0.32)	3.68 (0.39)
Word Sub	3.39 (0.43)	3.86 (0.49)	3.58 (0.35)	2.42 (0.48)
Zhu	2.78 (0.45)	2.59 (0.48)	2.82 (0.37)	2.93 (0.50)
RevILP	3.13 (0.36)	3.18 (0.45)	3.28 (0.32)	2.96 (0.39)
PBMT-R	3.47 (0.46)	3.83 (0.49)	3.71 (0.44)	2.88 (0.46)

Table 3.5: Mean scores assigned by human subjects, with the standard deviation between brackets

3.4.1 *Human judgements*

To test for significance we ran repeated measures analyses of variance with system (Simple Wikipedia, PBMT-R, Zhu, RevILP, word-substitution baseline) as the independent variable, and the three individual metrics as well as their combined mean as the dependent variables. Mauchly's test for sphericity was used to test for homogeneity of variance, and when this test was significant we applied a Greenhouse-Geisser correction on the degrees of freedom (for the purpose of readability we report the normal degrees of freedom in these cases). Planned pairwise comparisons were made with the Bonferroni method. Table 3.5 displays these results.

First, we consider the 3 metrics in isolation, beginning with fluency. We find that participants rated the fluency of the simplified sentences from the four systems and Simple Wikipedia differently, $F(4, 180) = 178.436, p < .001, \eta_p^2 = .799$. The word-substitution baseline, Simple Wikipedia and PBMT-R receive the highest scores (3.86, 3.84 and 3.83 respectively) and do not achieve significantly different scores on this dimension. All other pairwise comparisons are significant at $p < .001$. RevILP attains a score of 3.18, while the Zhu system achieves the lowest mean judgement score of 2.59.

Participants also rated the systems significantly differently on the adequacy scale, $F(4, 180) = 116.509, p < .001, \eta_p^2 = .721$. PBMT-R scores highest (3.71), followed by the word-substitution baseline (3.58), RevILP (3.28), and then by Simple Wikipedia (2.91) and the Zhu system (2.82). Simple Wikipedia and the Zhu system do not differ significantly, and all other pairwise comparisons are significant at $p < .001$. The low score of Simple Wikipedia indicates indirectly that the human editors of Simple Wikipedia texts often choose to deviate quite markedly from the meaning of the original text.

Key to the task of simplification are the human judgements of simplicity. Participants rated the simplicity of the output from the four systems and Simple Wikipedia differently, $F(4, 180) = 74.959, p < .001, \eta_p^2 = .625$. Simple Wikipedia scores highest (3.68) and the word substitution baseline scores lowest (2.42). Between them are the RevILP (2.96), Zhu (2.93) and PBMT-R (2.88) systems, which do not score significantly differently from each other. All other pairwise comparisons are significant at $p < .001$.

Finally we report on a combined score created by averaging over the fluency, adequacy and simplicity scores. In this way we get one score for each system indicating the quality of its output. Inspection of this score, displayed in the leftmost column of Table 3.5, reveals that the PBMT-R system and Simple Wikipedia score best (3.47 and 3.46 respectively), followed by the

Wikipedia	the judge ordered that chapman should receive psychiatric treatment in prison and sentenced him to twenty years to life , slightly less than the maximum possible of twenty-five years to life .
Simple Wikipedia	he was sentenced to twenty-five years to life in prison in 1981 .
Word-substitution baseline	the judge ordered that chapman should have psychiatric treatment in prison and sentenced him to twenty years to life , slightly less than the maximum possible of twenty-five years to life .
Zhu	the judge ordered that chapman should get psychiatric treatment . in prison and sentenced him to twenty years to life , less maximum possible of twenty-five years to life .
RevILP	the judge ordered that chapman should will get psychiatric treatment in prison . he sentenced him to twenty years to life to life .
PBMT-R	the judge ordered that chapman should get psychiatric treatment in prison and sentenced him to twenty years to life , a little bit less than the highest possible to twenty-five years to life .

Wikipedia	this array distributes data across multiple disks , but the array is seen by the computer user and operating system as one single disk .
Simple Wikipedia	the user will only see the logical disk.the operating system will not see the raid , it will see a single disk .
Word-substitution baseline	this table distributes data across more disks , but the array is seen by the user and computer operating system as a single disk .
Zhu	this array sells data across multiple disks but the array is seen . the computer user and operating as one disk .
RevILP	this array distributes data across many disks . the array is seen by the computer user and operating system as one single disk .
PBMT-R	this array distributes data across multiple disks , but the array is seen by the computer user and operating system as one single record .

Table 3.6: Example output of the simplification systems

word substitution baseline (3.39), which in turn scores higher than RevILP (3.13) and the Zhu system (2.78). We find that participants rated the systems significantly differently overall, $F(4, 180) = 98.880, p < .001, \eta_p^2 = .687$. All pairwise comparisons were statistically significant ($p < .01$), except the one between the PBMT-R system and Simple Wikipedia.

3.4.2 *Correlations*

Table 3.7 displays the correlations between the scores assigned by humans (fluency, adequacy and simplicity) and the automatic metrics (Flesch-Kincaid and BLEU). We see a significant correlation between fluency and adequacy (0.45), as well as between fluency and simplicity (0.24). There is a negative significant correlation between Flesch-Kincaid scores and simplicity (-0.45) while there is a positive significant correlation between Flesch-Kincaid and adequacy and fluency. The significant correlations between BLEU and simplicity (0.42) and fluency (0.26) are both in the positive direction. There is no significant correlation between BLEU and adequacy, indicating BLEU's relative weakness in assessing the semantic overlap between input and output. BLEU and Flesch-Kincaid do not show a significant correlation.

3.5 DISCUSSION

	adequacy	simplicity	Flesch-Kincaid	BLEU
fluency	0.45**	0.24*	0.42**	0.26**
adequacy		-0.19	0.40**	-0.14
simplicity			-0.45**	0.42**
Flesch-Kincaid				-0.11

Table 3.7: Pearson correlation between the different dimensions as assigned by humans and the automatic metrics. Scores marked * are significant at $p < .05$ and scores marked ** are significant at $p < .01$

We conclude that a phrase-based machine translation system with added dissimilarity-based re-ranking of the best ten output sentences can successfully be used to perform sentence simplification. Even though the system merely performs phrase-based machine translation and is not specifically geared towards simplification were it not for the dissimilarity-based re-ranking of the output, it performs not significantly differently from state-of-the-art

sentence simplification systems in terms of human-judged simplification. In terms of fluency and adequacy our system is judged to perform significantly better. From the relatively low average numbers of edits made by our system we can conclude that our system performs relatively small numbers of changes to the input, that still constitute sensible simplifications. It does not split sentences (which the Zhu and RevILP systems regularly do); it only rephrases phrases. Yet, it does this better than a word-substitution baseline, which can also be considered a conservative approach; this is reflected in the baseline's high fluency score (roughly equal to PBMT-R and Simple Wikipedia) and adequacy score (only slightly worse than PBMT-R).

The output of all systems, the original and the simplified version of an example sentence from the PWKP dataset is displayed in Table 3.6. The Simple Wikipedia sentences illustrate that significant portions of the original sentences may be dropped, and parts of the semantics of the original sentence discarded. We also see the Zhu and RevILP systems resorting to splitting the original sentence in two, leading to better Flesch-Kincaid scores. The word-substitution baseline changes 'receive' in 'have', while the PBMT-R system changes the same 'receive' in 'get', 'slightly' to 'a little bit', and 'maximum' to 'highest'.

In terms of automatic measures we see that the Zhu system scores particularly well on the Flesch-Kincaid metric, while the RevILP system and our PBMT-R system achieve the highest BLEU scores. We believe that for the evaluation of sentence simplification, BLEU is a more appropriate metric than Flesch-Kincaid or a similar readability metric, although it should be noted that BLEU was found only to correlate significantly with fluency, not with adequacy. While BLEU and NIST may be used with this in mind, readability metrics should be avoided altogether in our view. Where machine translation evaluation metrics such as BLEU take into account gold references, readability metrics only take into account characteristics of the sentence such as word length and sentence length, and ignore grammaticality or the semantic adequacy of the content of the output sentence, which BLEU is aimed to implicitly approximate by measuring overlap in n-grams. Arguably, readability metrics are best suited to be applied to texts that can be considered grammatical and meaningful, which is not necessarily true for the output of simplification algorithms. A disruptive example that would illustrate this point would be a system that would randomly split original sentences in two or more sequences, achieving considerably lower Flesch-Kincaid scores, yet damaging the grammaticality and semantic coherence of the original text, as is evidenced by the negative correlation for simplicity and positive correlations for fluency and adequacy in Table 3.7.

A valuable modification to the system would be a boost in the number of edits the system performs, while still producing grammatical and meaning-preserving output. Although the comparison against the Zhu system, which uses syntax-driven machine translation, shows no clear benefit for syntax-based machine translation, it may still be the case that approaches such as Hiero (Chiang et al., 2005) and Joshua (Li et al., 2009), enhanced by dissimilarity-based re-ranking, would improve over our current system. Furthermore, typical simplification operations such as sentence splitting and more radical syntax alterations or even document-level operations such as manipulations of the co-reference structure would be interesting to implement and test.

4

SENTENCE COMPRESSION

In this chapter we will discuss the task of sentence compression. We will present a memory-based approach that can perform compression by deletion (extractive compression) and a hybrid model that makes use of phrase-based machine translation that in addition to deletions can also perform compressions by paraphrasing longer source phrases into shorter target phrases (abstractive compression). Because no sufficiently large abstractive compression corpora exist, we train the phrase-based machine translation component of the hybrid model on simplification data. We will describe the extractive and abstractive systems and let human judges evaluate the output of these systems. Although in general we expect humans to evaluate abstractive compression more positive than extractive compression, abstractive compression is a task that is considerably more difficult than extractive compression, and there is no abstractive data available. We therefore expect the extractive approach to be a strong approach.

4.1 INTRODUCTION

Sentence compression can be defined as the task of producing a summary of a single sentence, with the goal of reducing the length of that sentence. The shortened version of the sentence should still be grammatical and retain the most important information contained in the source sentence. Consider the following sentences:

(4.1) " I can not guarantee that there will be no more coup attempts , " said the armed forces spokesman , Brigadier-General Oscar Florendo .

(4.2) " I can not guarantee that there will be no more coup attempts , " said the armed forces spokesman .

(4.3) The armed forces spokesman could not guarantee there will be no more coup attempts .

Here, sentence (4.1) is an original, uncompressed sentence. Sentences (4.2) and (4.3) are two human generated compressions, obtained by using different strategies. In (4.2), words from the original sentence have been deleted to create the compressed sentence. In (4.3), in addition to the deletion of words, phrases from the original sentence have been paraphrased into shorter phrases to create an even shorter sentence. In this chapter we explore automatic techniques for sentence compression, which utilize the strategies displayed in (4.2) and (4.3).

The task of sentence compression is quite similar to the task of summarization, albeit scaled down to the sentence level as opposed to the document level (Knight and Marcu, 2002). Sentence compression is also a crucial part of summarization: after the relevant sentences from the document are selected to form the summary, redundant information from these sentences can be removed by using sentence compression to further reduce the length of the summary (Lin, 2003; Jing and McKeown, 2000). Other examples of applications of automatic sentence compression are subtitle generation from spoken transcripts (Vandeghinste and Pan, 2004; Daelemans et al., 2004), and displaying text on devices with small screens such as PDA's or mobile phones (Corston-Oliver, 2001).

The development of sentence compression models started in support of summarization, where a summary was realized by deleting a subset of the text. This approach generates an extract of the text and is called extractive summarization. Much of the prior work has focused on extractive approaches to sentence compression (Knight and Marcu, 2002). Given a source sentence containing a set of words, this approach tries to find a subset of

these words that can be dropped to create a new, shorter sentence that is still grammatical and contains the most important information. More formally, the aim is to shorten a sentence $x = x_1, x_2, ..., x_n$ into a substring $y = y_1, y_2, ..., y_m$ where all words in y also occur in x in the same order and $m < n$. Sentence (4.2) is an example. Different techniques have been used for extractive sentence compression, ranging from the noisy-channel model (Knight and Marcu, 2002; Turner and Charniak, 2005), large-margin learning (McDonald, 2006; Cohn and Lapata, 2007) to Integer Linear Programming (Clarke and Lapata, 2008). (Marsi et al., 2010) characterize these approaches in terms of two assumptions: (1) only word deletions are allowed and (2) the word order is fixed. They argue that these constraints rule out more complicated operations such as reordering, substitution and insertion, and reduce the sentence compression task to a word deletion task. This does not model human sentence compression accurately, as humans tend to use the more complicated operations when transforming sentences into a summary (Jing and McKeown, 2000) as is displayed in example (4.3). This approach is called abstractive summarization.

In this chapter, we will investigate a memory-based approach to the task of sentence compression by deletion, and an extension that makes use of phrase-based machine translation to allow for paraphrasing operations that shorten the sentence beyond just deletions. We will first describe the memory-based deletion model that can delete phrases to produce shorter sentences. We then describe the hybrid model that combines the memory-based deletion model with phrase-based machine translation in order to delete phrases, but also paraphrase parts of the sentence to produce a shorter sentence. In the hybrid system, the extra compression provided by the paraphrasing model allows the memory-based deletion model to perform fewer deletions and yet end up with a similar compression rate as the deletion-only model. Our hypothesis is that this will result in more natural and structurally better compressions. The paraphrase-based approach has been proven to work for the related tasks of paraphrasing (Quirk et al., 2004; Zhao et al., 2009; Wubben et al., 2010) and sentence simplification (Coster and Kauchak, 2011; Wubben et al., 2012). To test our hypothesis we will evaluate both approaches by having humans judge the output of both systems at roughly the same compression rates.

4.1.1 *Related work*

(Knight and Marcu, 2002) used the Ziff-Davis corpus to extract a selection of sentences paired with compressions. The Ziff-Davis corpus is a collection

of news articles about technological products. For each article there is an abstract. If a sentence from the abstract is a subsequence of words from a sentence in the article, it is added to the sentence compression collection. From the collection of over 4,000 articles, Knight and Marcu extract 1067 of these sentence pairs. This low number can be explained by the fact that many sentences in the abstract do not meet the criteria set by Knight and Marcu. Parts of the sentences in the articles are often paraphrased to generate the abstract, or sentences are combined to form one sentence in the abstract. These examples do not fit the definition of sentence compression as a deletion task and have been left out of their set. In later research, these non-extractive compressions have also largely been ignored. (Knight and Marcu, 2002) propose two models to generate a short sentence by deleting a subset of words: the decision tree model and the noisy channel model, both based on a synchronous context free grammar. The decision tree model decomposes the rewriting process into a sequence of shift-reduce-drop actions that lead to deletion. A decision tree is learned this way to incrementally transform the source tree into the compressed target tree. The noisy channel model states that a target long sentence T originated from a short source sentence S and at some point words were added to generate sentence T. The task is now to find the most likely short sentence S for a given T. This can be defined as maximizing the following equation:

$$P(T, S) = P(T)P(T|S)$$

The probabilities for $P(T|S)$ can be estimated by using the frequencies of these operations on the parse trees of the sentences in the training data. Turner and Charniak (2005) and Galley and McKeown (2007) built upon this model reporting improved results.

McDonald (2006) developed a system using large-margin online learning combined with a decoding algorithm that searches the compression space to produce a compressed sentence. He uses a rich feature set of unigrams, bigrams and part of speech tags in the original sentence and dropped words and phrases from the original uncompressed sentence, as well as soft syntactic features from the dependency and parse trees of each sentence. Discriminative learning is used to combine the features and weight their contribution to a successful compression.

Cohn and Lapata (2007) cast the sentence compression problem as a tree-to-tree rewriting task. For this task, they train a synchronous tree substitution grammar, which dictates the space of all possible rewrites. By using discriminative training, a weight is assigned to each grammar rule. These grammar rules are then used to generate compressions by a decoder. Cohn

and Lapata achieve state-of-the-art results applying this method to extractive sentence compression.

One application for which sentence compression has been investigated is automatic subtitling (Vandeghinste and Pan, 2004). Vandeghinste and Pan (2004) describe a sentence compression system trained on Dutch transcript - subtitle pairs from television programmes. They use tagging, shallow parsing, subordinate clause detection and rules to generate compressions in Dutch. For each chunk or clause they decide whether to delete it, keep it or reduce it. Reduction is done by keeping only the head of noun compounds.

In contrast to the large body of work on extractive sentence compression, work on abstractive sentence compression is relatively sparse. Cohn et al. (2008) propose an abstractive sentence compression method based on a parse tree transduction grammar and Integer Linear Programming. To train their model, they compiled a corpus from 30 newspaper articles from the British National Corpus (BNC) and the American News Text corpus, containing a total of 575 sentences. An annotator produced a compression for each sentence. The annotator was instructed to create a compression by using any rewrite operation: deletion, insertion, reordering or substitution. A transduction grammar is then learned from the pairs of sentences, creating an extractive model that can transform source sentences into shorter target sentences by deletion. For their abstractive model, the grammar that is extracted is augmented with paraphrasing rules obtained from a pivoting approach to a bilingual corpus (Bannard and Burch, 2005). They show that the abstractive model outperforms the extractive model on their dataset. Another paraphrasing approach is the one by Zhao et al. (2009). They use a paraphrase model inspired by phrase-based machine translation that can be tailored to perform different tasks. One of these tasks is sentence compression (Zhao et al., 2009).

4.1.2 *This study*

In this study we propose a simple memory-based deletion model for sentence compression, which is trained to delete parts of a sentence to produce a compressed sentence. We compare this approach to a model that combines a memory-based deletion model with an additional paraphrasing component to allow for abstractive compression. Our aim is twofold. First, we want to investigate if it is possible to model the sentence compression task as a deletion task, where possible deletions are represented by delete operations on parse trees and stored in memory. Our second aim is to investigate if it is feasible to improve this extractive method by combining it with a para-

phrase component which has been shown to be successful in related tasks of paraphrasing and sentence simplification. Our method is to evaluate the output of the extractive and abstractive model along with a human produced abstractive compression automatically and by letting human judges judge the output of the models on two different dimensions. The models are also evaluated on compression character rate, rather than token compression rate. Character compression rates are important, because in addition to deleting tokens we can also compress sentences by paraphrasing longer words into shorter words, ending up with a shorter sentence in terms of characters, but not necessarily in terms of tokens. Character counts are also vital to various applications, such as Twitter, which allows messages of up to 140 characters. We aim to keep the compression character rates for the different systems similar, in order to be able to compare the performance of the systems in a meaningful way (Napoles et al., 2011). The compression rate over characters is calculated as follows:

$$CR = \frac{Characters_{target}}{Characters_{source}}$$

Because of the versatility of our deletion model, we can adjust the character compression rate as we will demonstrate in the following section where we discuss the memory-based deletion model. In the subsequent section we will discuss the hybrid model that combines the deletion based approach with a monolingual machine translation component. Following that, we will discuss an experiment conducted with human judges, and we will end the chapter with a critical discussion of the results.

4.2 SENTENCE COMPRESSION MODELS

4.2.1 *Extractive model*

An extractive compression model generates a summary of a sentence by removing words from the sentence. Consider the following sentences:

(4.4)
1. lebanese parliamentary sessions have to be open to the public .
2. parliamentary sessions have to be open .

The first sentence in this example is the source sentence and the second sentence is the compression which has been generated by our extractive system by deleting a subset of words in the source sentence. The memory-based

deletion system creates this extractive summary by removing non-terminal nodes from the parse tree of the sentence. The system is trained on the manually annotated extractive sentence compression corpus developed by (Clarke and Lapata, 2008). We use the written part of the corpus, containing pairs of original and compressed sentences from news articles. The corpus is tokenized and lowercased. Lines consisting of multiple sentences are removed, because they do not fit in the sentence compression model and cause the parser to split the line into multiple lines.

Of the 1422 remaining pairs of sentences, the original sentences are then parsed using the Stanford parser [1] (Klein and Manning, 2003). For each non-terminal node in the parse of the uncompressed sentence, we then check whether the annotator deleted it to produce the compressed version of the sentence. Each highest possible node in the parse tree that is deleted then gets assigned the class "delete". All the children nodes of the "delete" node receive the class "delete implied" and each of the children of "delete implied" nodes get the class "delete implied" as well. All other nodes are assigned the class "keep". The class for each node is stored in memory along with a selection of features, such as the syntactic category of the node, the syntactic category of the parent node, the depth of the tree at which the node is located and the length of the phrase contained within the node. All features that we use are displayed in Table 4.1 along with the number of different values of these features in the training set and the information gain ratio, which is used to weigh the features. We use the syntactic category of the node, the parent rule and the parent rule combined with the category of the node as features, as well as the rule of the child node and the depth at which the node is found (counted from the top node). In addition to these syntactic features, we also use as features the length of the node measured in words, the first word of the node and all the words in the node. This is exemplified in Table 4.2.

We split the instance base into a training set and a development set. The training set is down-sampled so that the number of "keep" instances is no bigger than the number of "delete" and "delete implied" combined. We end up with 33,274 training instances. Note that half of these instances are of the "keep" category.

With these training instances we are able to train a model to classify new instances. Our model makes use of memory-based learning (MBL) for classification. MBL stores feature representations of training instances in memory

1 version 2.0.1

feature	number of unique values	Information Gain Ratio
category	66	0.042
parent rule	1485	0.047
parent rule + category	3815	0.055
child rule	6191	0.057
depth	38	0.020
length	64	0.047
first word	4756	0.048
all words	14368	0.070

Table 4.1: Features with examples, number of unique values of these features and information gain ratio

without abstraction and classifies unseen instances by matching their feature representation to all instances in memory, finding the most similar instances. The class of the new instance is then extrapolated from the most similar instance(s) in memory. Because the task of sentence compression and many other Natural Language Processing tasks contain many exceptions and low-frequency instances, it can be argued that memory-based learning is at an advantage in solving these problems compared to algorithms that abstract from the instances (Daelemans, 1999). The learning algorithm our system uses is the IB1 classifier as implemented in TiMBL (version 6.3.2) (Daelemans et al., 2009). IB1 is a supervised decision-tree-based implementation of the k-nearest neighbor algorithm for learning classification tasks (Aha et al., 1991). We use TiMBL with default settings, while only varying k. By increasing the number of neighbors (k), we are more likely to select "keep" nodes, as this class is equally frequent as both other classes ("delete" and "delete implied") combined. This means that k acts as a means to control compression rate: an increase of k will result in an increase of compression rate, meaning a less aggressive compression. The effect of varying k on compression rate and F-score can be witnessed in Figure 4.1. The reported F-score is the micro F-score calculated over precision and recall. Upon inspection of the data in Figure 4.1 we set $k = 3$ for the final deletion mode, because the compression

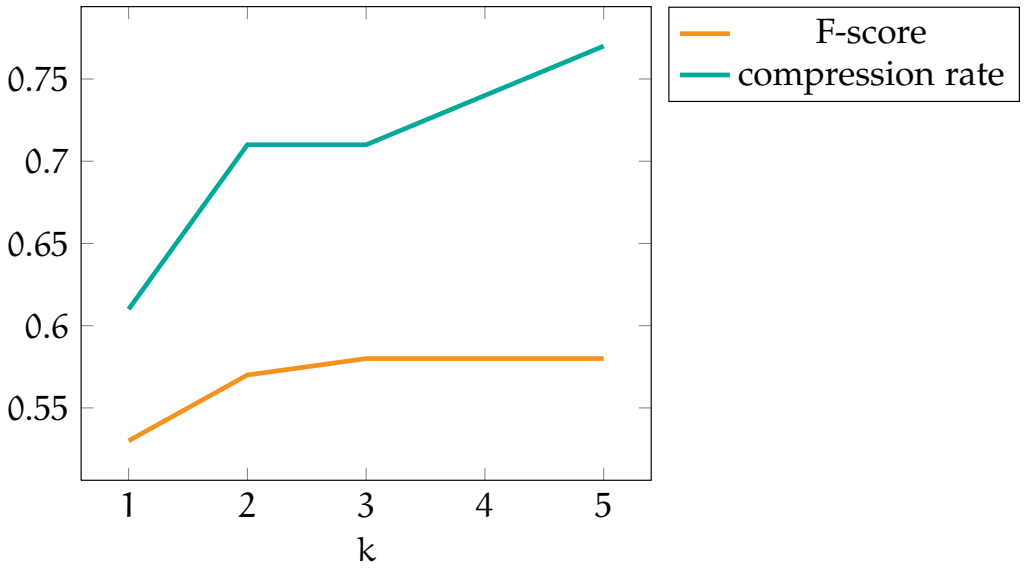


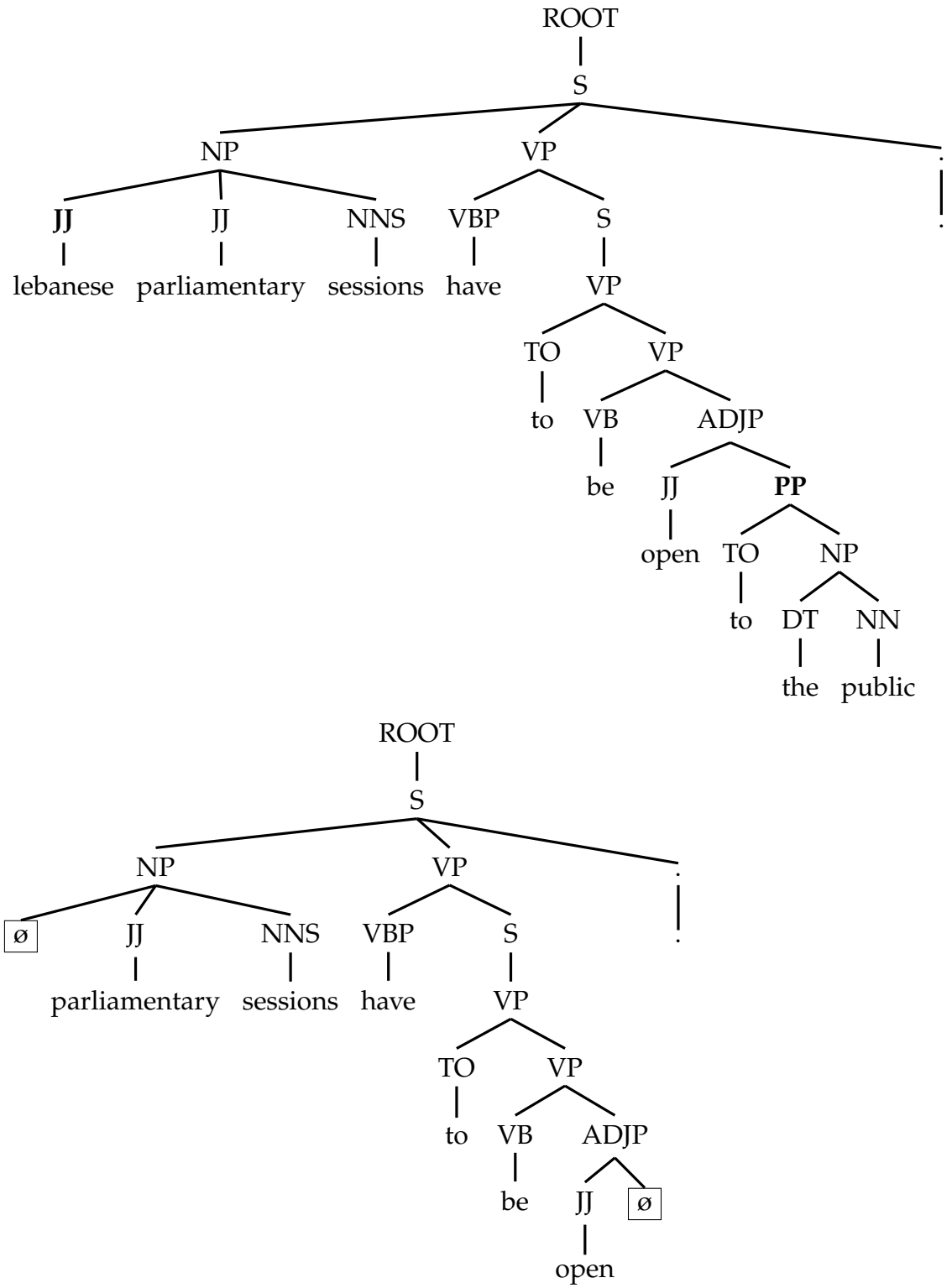
Figure 4.1: F-score for the classifier and compression rate when varying k

rate is quite low while the F-score is quite high. Each node that is labelled as "delete" will be deleted to produce a compressed sentence.

Our system takes the following steps in order to generate an extractive summary of a sentence: first, the sentence is parsed using the Stanford parser. Then, for each node we collect a number of features that describe this node. Table 4.2 is an example of the instance“to the public”. We run the memory-based classifier which assigns a class to each instance. In this example, the JJ node containing “lebanese” and the PP node containing “to the public” are classified as “delete”. Finally, we delete all words or phrases belonging to nodes marked “delete” to generate the output sentence. This process is illustrated in Figure 4.2. This model can be extended to incorporate the “delete implied” class in its decision to delete or keep a subtree to ensure a more conservative but also a more precise compression. We leave that extension for later studies.

4.2.2 *Abstractive model*

For the hybrid abstractive model we relax the deletion model by setting $k = 4$ and have the paraphrasing component take care of the rest of the compression. Figure 4.1 shows that the F-score of the memory-based classifier remains similar when setting $k = 4$, while the compression rate goes up. This means the classifier is less strict, resulting in a reduced removal of



source:	lebanese	parliamentary sessions have to be open	to the public	.
target:	ø	parliamentary sessions have to be open	ø	.

Figure 4.2: Example deletion in the extractive model. The boxed nodes of the parse are selected for deletion by the classifier, resulting in a deletion of the corresponding phrases.

feature	example
category	PP
parent rule	ADJP=>JJ PP
parent rule + category	ADJP=>JJ PP*
child rule	PP=>TO NP
depth	8
length	3
first word	to
all words	to the public

Table 4.2: Examples of features for the instance "to the public"

information. The remaining part of the compression is then done by paraphrasing longer phrases with shorter phrases. In order to make our compressing paraphrasing component work, we need a large aligned abstractive compression corpus. Because no sufficiently large corpora exists, we instead use a simplification corpus. The corpus we use is the PWKP corpus by Zhu et al. (2010). This corpus consists of aligned sentences from articles in English Wikipedia and English Simple Wikipedia. The PWKP dataset consists of 108,016 pairs of aligned lines from 65,133 English Wikipedia and English Simple Wikipedia articles. These articles were paired by following the "interlanguage link"[2]. TF*IDF at the sentence level was used to align the sentences in the different articles (Nelken and Shieber, 2006). This dataset was originally developed for sentence simplification. As was demonstrated in Chapter 3, the use of this dataset for the phrase-based machine translation approach to sentence simplification is a viable option. Although simplification does not necessarily mean compression, in many cases simplified sentences are in fact compressed as well. (Zhu et al., 2010) report the average sentence length of the English Wikipedia sentences in the PWKP corpus to be 25.01 tokens, and the English Simple Wikipedia to be 20.87 tokens. The average token length is 5.06 for English Wikipedia versus 4.89 in Simple English Wikipedia. Examples of compression in Simple Wikipedia can be observed in Table 4.3.

2 http://en.wikipedia.org/wiki/Help:Interlanguage_links

English Wikipedia:	Peter "Pete" Doherty (born 12 March 1979) is an English musician, writer, actor, poet and artist. He is best known musically for being co-frontman of The Libertines, which he reformed with Carl Barat in 2010. His other musical project is indie band Babyshambles.
Simple Wikipedia:	Peter Doherty (born 12 March 1979) is an English musician. He is in the popular rock band The Libertines. He is also in a band called Babyshambles.
English Wikipedia:	Owls are a group of birds that belong to the order Strigiformes, constituting 200 extant bird of prey species. Most are solitary and nocturnal, with some exceptions (e.g., the Northern Hawk Owl). Owls hunt mostly small mammals, insects, and other birds, although a few species specialize in hunting fish.
Simple Wikipedia:	Owls are birds in the order Strigiformes. Most of them are solitary and nocturnal. They feed on small mammals, insects, and other birds, and a few species like to eat fish as well.

Table 4.3: Examples of compression in sentences from articles from normal English Wikipedia and Simple English Wikipedia.

We use the Moses software to train a PBMT model (Koehn et al., 2007). A statistical machine translation model normally finds a best translation $\tilde{E}$ of a text in language F for a text in language E by combining a translation model $P(F|E)$ with a language model $P(E)$:

$$\tilde{E} = \arg\max_{E \in E^*} P(F|E)P(E)$$

The sentence F is divided into a sequence of I phrases during decoding in phrase-based machine translation. Each source phrase is then translated into a target phrase to form sentence E. These phrases may be reordered. Machine translation is described in more detail in Chapter 1.

The GIZA++ statistical alignment package is used to perform the word alignments, which are later combined into phrase alignments in the Moses pipeline (Och and Ney, 2003) to build the paraphrase model. GIZA++ implements IBM Models 1 to 5 and an HMM word alignment model to find

statistically motivated alignments between words. We first tokenize our data. We then lowercase all data and use all Simple Wikipedia sentences from the PWKP dataset to train an n-gram language model with the SRILM toolkit (Stolcke, 2002). Then we invoke the GIZA++ aligner using the training pairs. We run GIZA++ with standard settings and we perform no optimization. We then prune the resulting phrase table, keeping only phrases that have a translation that is shorter or of equal length, measured in characters.

We use the Moses decoder to generate compressions for the sentences in our test data. To expand the functionality of Moses in the intended direction we perform post-hoc re-ranking on the output based on the character compression rate relative to the input sentence. We do this to select output that is as short as possible; at the same time, we base our re-ranking on a top-n of output candidates according to Moses, with a small n, to ensure that the quality of the output in terms of fluency and importance is also controlled for. Setting $n = 10$, for each source sentence we re-rank the ten best sentences as scored by the decoder according to the compression factor measured in characters between input and output sentence. In case of a tie in compression factor, we select the sequence with the better decoder score. When Moses is unable to generate ten different sentences, we select from the lower number of outputs.

Two example source sentences with generated compressions by humans and the two systems are displayed in Table 4.4. In the first sentence, we see that the extractive and abstractive model both delete the same parts of the sentence. In addition, the abstractive model paraphrases “defence establishments” incorrectly as “free companies”. In the second sentence, we see that the abstractive model does not perform deletion, but only compresses the sentence by paraphrasing “expenditure” into “us”. The replacement of “exercise” by "practise" does not yield any compression. In contrast, the extractive model erroneously deletes “said”. We can also see that the human references are also not perfect, in this example “spokeswoman” is replaced by the erroneous “spokewoman”.

4.3 EVALUATION

To evaluate the output of our systems we collect automatic scores (ROUGE scores and character compression rates) as well as human judgements on two different dimensions (fluency and importance).

source	cutbacks in local defence establishments is also a factor in some constituencies .
human referent	cutbacks in local defence establishments is also a factor .
extractive model	cutbacks in defence establishments is a factor .
abstractive model	cutbacks in free companies is a factor .
source	the family expenditure surveys will be a part of the exercise , " a spokeswoman said .
human referent	the family expenditure surveys will be included , " a spokewoman said .
extractive model	the family expenditure surveys will be a part of the exercise , " a spokeswoman .
abstractive model	the family use surveys will be a part of the practice , " a spokeswoman said .

Table 4.4: Example output

4.3.1 *Participants*

The participants in this evaluation study were 19 students of Tilburg University, who participated for partial course credits. All participants were native speakers of Dutch, and all were proficient in English, having taken a course on Academic English at University level.

4.3.2 *Materials*

To test our systems we need a data set containing abstractive compressions of sentences. We use the test set developed by (Cohn and Lapata, 2008). This test set is part of a corpus of 30 news articles from the British National Corpus and the American News Text corpus for which they obtained manual compressions from human annotators. These compressions were generated by two annotators who were asked to use any appropriate rewriting operation (deletion, reordering, substitution, addition) to compress the sentence. We randomly pick 35 source sentences, resulting in 35 clusters of one source sentence and three compressed sentences, as generated by humans, the Ex-

tractive model (i.e. the memory-based deletion model) and the Abstractive model (i.e. the hybrid phrase-based machine translation model). The training material for the abstractive model is from another domain than the test set, so the performance of the abstractive model will probably be lower than when we would use training and testing materials from the same domain. Since no sufficiently large news corpus of abstractive compressions exists, using out of domain data is the only choice.

4.3.3 *Procedure*

The participants were told that they participated in the evaluation of a system that could compress sentences, and that they would see one source sentence and three automatically compressed versions of that sentence. They were not informed of the fact that we were actually evaluating two different systems and a human referent. Following earlier evaluation studies (Knight and Marcu, 2002; Clarke and Lapata, 2008; Cohn and Lapata, 2008) we asked participants to evaluate fluency and importance of the target compressions on a five point Likert scale. Fluency was defined in the instructions as the extent to which a sentence is proper, grammatical English. Importance was defined as the extent to which the sentence has retained the important information from the source sentence. The order in which the clusters had to be judged was randomized and the order of the output of the various systems was randomized as well.

4.4 RESULTS

We will first report on the scores assigned by the automatic measures, followed by human judgements and the correlations between automatic and human assigned scores.

4.4.1 *Automatic measures*

As is witnessed in Table 4.5, the two systems achieve a similar character compression ratio of 0.74. This is not quite as low as the human compression ratio, which is 0.58. For the automatic evaluation we use ROUGE (Lin, 2004) (Recall-Oriented Understudy for Gisting Evaluation). ROUGE is a set of metrics widely used in the evaluation of automatic summarization. It compares an automatically produced summary against one or more human produced reference summaries. In the context of summarization of news articles, ROUGE has been shown to correlate with human evaluations in re-

gards of how well the contents match (Lin, 2004). It has also been used to evaluate Sentence Compression (Zajic et al., 2007; Liu and Liu, 2009). We use the F-measure for both ROUGE-1 (measuring only unigrams) and ROUGE-SU4, which measures skip-bigrams with a maximum gap of 4 tokens. For both ROUGE-1 and ROUGE-SU4 the Extractive model receives the highest scores (0.56 and 0.34 respectively).

system	compression rate	ROUGE-1	ROUGE-SU4
human referent	0.58 (0.16)	-	-
extractive	0.74 (0.17)	0.56 (0.19)	0.34 (0.22)
abstractive	0.74 (0.13)	0.53 (0.18)	0.29 (0.18)

Table 4.5: Character compression rates of the different systems and the human referent and ROUGE scores with standard deviations between brackets

4.4.2 *Human judgements*

In this section we report on the human judgements. To test for significance of the difference in judgements of the different systems we ran repeated measures analyses of variance with system (human referent, Extractive model, Abstractive model) as the independent variable, and the three individual metrics as well as their combined mean as the dependent variables. Mauchly's test for sphericity was used to test for homogeneity of variance, and when this test was significant we applied a Greenhouse-Geisser correction on the degrees of freedom (for the purpose of readability we report the normal degrees of freedom in these cases). Planned pairwise comparisons were made with the Bonferroni method. Table 4.6 displays these results.

system	overall	fluency	importance
human referent	3.61(0.47)	3.98 (0.53)	3.24 (0.46)
extractive	3.39 (0.33)	3.29 (0.37)	3.50 (0.35)
abstractive	3.10 (0.38)	2.94 (0.39)	3.26 (0.45)

Table 4.6: Mean scores assigned by human subjects, with the standard deviation between brackets

First, we consider the 2 metrics in isolation, beginning with fluency. We find that participants rated the fluency of the compressed sentences from

the two systems and the human referent differently, $F(2, 36) = 65.186, p < .001, \eta_p^2 = .784$. All other pairwise comparisons are significant at $p < .001$. The human referent receives the highest score (3.98), followed by the extractive model (3.29) and the abstractive model scores lowest (2.94).

The participants also rated the systems significantly differently on the importance scale, $F(2, 36) = 4.575, p < .001, \eta_p^2 = .203$. The Extractive model scores significantly higher (3.50) than the Abstractive model (3.26) at $p < .001$. The difference in score between the Extractive model and the human referent approaches significance (3.24) at $p < .07$.

Finally we report on a combined score that is created by averaging over the fluency and importance scores. Inspection of this score, displayed in the leftmost column of Table 4.6, reveals that the referent and the extractive model score best (3.61 and 3.39 respectively) and not significantly differently, followed by the significantly lower scoring abstractive model (3.10) ($p < .001$).

4.4.3 *Comparison with Cohn and Lapata (2008)*

Unfortunately we were unable to acquire the output from Cohn and Lapata's systems. Therefore, we can only compare our results to their results, while bearing in mind the caveat that their method, test subjects and sample of sentences all differ from ours. Cohn and Lapata evaluate on 30 randomly chosen sentences from the test set. They collected fluency (in their paper called Grammaticality) and importance ratings from 19 judges over the internet for the extractive model, the abstractive model and the human referent. Their reported results are in Table 4.7. In general, their extractive model receives lower scores than our extractive model, while compressing less (0.83 is the character compression rate of their system, compared to 0.74 of ours). Their abstractive model gets higher fluency scores but lower importance scores than our abstractive model, but still compressing less (0.79 versus 0.74).

system	fluency	importance	compression rate
human referent	4.51	4.02	0.58
extractive	3.10	2.43	0.83
abstractive	3.38	2.85	0.79

Table 4.7: Results reported by Cohn and Lapata (2008)

4.4.4 *Correlations*

To gain some insight in the effect of compression rate on automatic and human judgements, we evaluate the correlations between the various variables. For this, we use the mean scores for only the generated sentences. When we consider the Pearson correlation between the different mean scores for the two systems, we see that the ROUGE metrics show no significant correlation with human judgements or character compression rates. We do see significant correlations between importance and fluency (0.56) , importance and compression rate (0.68) and fluency and compression rate (0.24). The high correlation between importance and compression rate supports the notion that the more a sentence is compressed, the more important information is lost in the process.

	fluency	compression rate	ROUGE-1	ROUGE-SU4
importance	0.56**	0.68**	0.09	0.11
fluency		0.24*	0.14	0.13
compression rate			0.11	0.16
ROUGE-1				0.93**

Table 4.8: Pearson correlation between the different dimensions as assigned by humans and the automatic metrics. Scores marked * are significant at $p < .05$ and scores marked ** are significant at $p < .01$

4.5 DISCUSSION

We have applied an extractive memory-based deletion model and an abstractive model that uses memory-based deletion plus phrase-based monolingual machine translation to the task of sentence compression. We have measured character compression rates for the systems and the referent and we have reported ROUGE-1 and ROUGE-SU4 scores for the generated compressions. In addition, we have evaluated our systems by having humans judge the output of the systems and a human authored compression on both fluency and importance. In terms of both automatic and human assigned scores, the abstractive version of the model fails to achieve higher scores than the extractive model on similar compression rates. Additionally, we have measured character compression rates for the systems and the referent and we

have reported ROUGE-1 and ROUGE-SU4 scores for the generated compressions.

We think that the fact that the abstractive model performs worse than the extractive model can be attributed to various factors. First, the extractive memory-based deletion model is a strong model: it can effectively create compressions at relatively low compression rates. Although it can be argued that compression is not a deletion only task, deletion remains the easiest operation to automatically perform and is also the most effective in terms of character compression counts. Secondly, no sufficiently large corpora exist for abstractive sentence compression. Instead, we opted to train our abstractive phrase-based machine translation paraphrasing component on simplification data, which might not be ideal for the compression task. We opted to prune the resulting phrase-table, to only keep the entries that actually compress. It is worth mentioning that the test set is from another domain than the training data: news messages versus Wikipedia data. It should also be taken into account that the input on which the paraphrasing system works has already been compressed by the extractive component. This means that the parts that can most easily be compressed have most likely already been deleted by the extraction model. We also see that the human referent does not score highest on importance. This can be explained by the fact that the human referent has a much lower compression rate than the systems (0.58 versus 0.74). This means that inevitably more information will be lost.

The deletion-based model can be expanded by adding a decoding process that selects the most appropriate deletions through the use of a language model and statistics generated by the classifier to generate better formed sentences. We have several tools at our disposal to tweak the compression process. On the one hand, we can use the "delete implied" class in the output of our classifier to generate more conservative but better informed compressions. On the other hand, we can approach human levels of compression rate by lowering k. How these variables impact compression quality is an interesting question.

We think the phrase-based machine translation paraphrasing component can be improved as well. One thing that would be needed for that is a large parallel abstractive sentence compression corpus. A possible venue for future exploration is the use of edit histories in Wikipedia for the purpose of creating such a corpus. In the process of editing Wikipedia many editors add, rephrase or remove information. Rephrasings that shorten sentences might be harvested and used in sentence compression.

5

LANGUAGE TRANSFORMATION

In this chapter we investigate language transformation. Language transformation can be defined as translating between diachronically distinct language variants. We investigate the transformation of Middle Dutch into Modern Dutch by means of machine translation. For diachronic language transformation we have to rely on parallel data that has been made available consisting of paired sentences from the two diachronic variants of the language. This means generally not much data is available. We tackle this problem by making use of the characteristics of the data. We demonstrate that by using character overlap the performance of the machine translation process can be improved for this task.

THIS CHAPTER IS BASED ON: Wubben, S., van den Bosch, A.P.J., & Krahmer, E.J. *Language transformation using character overlap and machine translation* (submitted for publication)

5.1 INTRODUCTION

In this chapter we describe a variant of paraphrasing that is more similar to machine translation than the approaches described in previous chapters. We aim to develop a system to translate between diachronically distinct language variants. For research into history, historical linguistics and diachronic language change, historical texts are of great value. Specifically from earlier periods, texts are often the only forms of information that have been preserved. One problem that arises when studying these texts is the difference between the language the text is written in and the modern variant that the researchers who want to study the texts know and speak themselves. It takes a great deal of deciphering and interpretation to be able to grasp these texts. Our aim is to facilitate laymen in studying medieval texts by attempting to generate literal translations of the sentences in the text into modern language. In particular we focus on the task of translating Middle Dutch to Modern Dutch. The transformation between language variants, either synchronically or diachronically, is more a regular translation task, just as it is often impossible to identify two languages as language variants or two different languages. This task differs from other paraphrasing tasks in that we are not interested in dissimilarity, instead we merely wish to give a likely translation in Modern Dutch of a Middle Dutch text. For paraphrasing it is essential to generate from each source sentence a target sentence in the same language that differs sufficiently from the source sentence while still carrying the same meaning. The transformation between language variants, either synchronically or diachronically, is more a regular translation task, just as it is often impossible to identify two languages as language variants or two different languages.

Middle Dutch can be defines as a collection of closely related West Germanic dialects that were spoken and written between 1150 and 1500 in the area that is now defined as the Netherlands and parts of Belgium. One of the factors that makes Middle Dutch difficult to read is the fact that at the time no overarching standard language existed. Modern Dutch is defined as Dutch as spoken from 1500. The variant we investigate is contemporary Dutch. Another difference with regular paraphrasing is the amount of parallel data available. The amount of parallel data for the variant pair Middle Dutch - Modern Dutch is several orders of magnitude smaller than our parallel headline data for the paraphrase generation task described in Chapter 2.

We do expect many etymologically related words to show a certain amount of character overlap between the Middle and Modern variants. An example of the data is given below, from the work 'Van den vos Reynaerde', part of

the Comburg manuscript that was written between 1380-1425. Here, (5.1) is the original text, (5.2) is a modern translation in Dutch by Walter Verniers and (5.3) is a translation in English added for clarity.

(5.1) "Doe al dat hof versamet was,
Was daer niemen, sonder die das,
Hine hadde te claghene over Reynaerde,
Den fellen metten grijsen baerde.

(5.2) "Toen iedereen verzameld was,
was er niemand -behalve de das-
die niet kwam klagen over Reynaert,
die deugniet met zijn grijze baard."

(5.3) "When everyone was gathered,
there was noone -except the badger-
who did not complain about Reynaert,
that rascal with his grey beard."

We can observe that although the texts (5.1) and (5.2) are quite different, there is a large amount of character overlap in the words that are used. Our aim is to use this character overlap to compensate for the lower amount of data that is available. We compare three different approaches to translate Middle Dutch into Modern Dutch: a standard Phrase-Based machine translation approach, a PBMT approach with additional preprocessing based on Needleman-Wunsch sequence alignment, and a character bigram based PBMT approach. The PBMT approach with preprocessing identifies likely translations based on character overlap and adds them as a dictionary to improve the statistical alignment process. The PBMT approach based on character bigrams rather than translating words, transliterates character bigrams and in this way improves the transformation process. We demonstrate that these two approaches outperform standard PBMT in this task, and that the PBMT transliteration approach based on character bigrams performs best.

5.1.1 *Related work*

Language transformation by machine translation within a language is a task that has not been studied extensively before. Related work is the study by Xu et al. (2012). They evaluate paraphrase systems that attempt to paraphrase a specific style of writing into another style. The plays of William Shakespeare and the modern translations of these works are used in this study. They

show that their models outperform baselines based on dictionaries and out-of-domain parallel text.

Work that is slightly comparable is the work by Zeldes (2007), who extrapolates correspondences in a small parallel corpus taken from the Modern and Middle Polish Bible. The correspondences are extracted using machine translation with the aim of deriving historical grammar and lexical items. A larger amount of work has been published about spelling normalization of historical texts. Baron and Rayson (2008) developed tools for research in Early Modern English. Their tool, VARD 2, finds candidate modern form replacements for spelling variants in historical texts. It makes use of a dictionary and a list of spelling rules. By plugging in other dictionaries and spelling rules, the tool can be adapted for other tasks. Kestemont et al. (2010) describe a machine learning approach to normalize the spelling in Middle Dutch Text from the 12th century. They do this by converting the historical spelling variants to single canonical (present-day) lemmas. Memory-based learning is used to learn intra-lemma spelling variation. Although these approaches normalize the text, they do not provide a translation.

More work has been done in the area of translating between closely related languages and dealing with data sparsity that occurs within these language pairs (Hajič et al., 2000; Van Huyssteen and Pilon, 2009). Koehn et al. (2003) have shown that there is a direct negative correlation between the size of the vocabulary of a language and the accuracy of the translation. Alignment models are directly affected by data sparsity. Uncommon words are more likely to be aligned incorrectly to other words or, even worse, to large segments of words (Och and Ney, 2003). Out of vocabulary (OOV) words also pose a problem in the translation process, as systems are unable to provide translations for these words. A standard heuristic is to project them into the translated sentence untranslated.

Various solutions to data sparsity have been studied, among them the use of part-of-speech tags, suffixes and word stems to normalize words (Popovic and Ney, 2004; De Gispert and Marino, 2006), the treatment of compound words in translation (Koehn and Knight, 2003), transliteration of names and named entities, and advanced models that combine transliteration and translation (Kondrak et al., 2003; Finch et al., 2012) or learning unknown words by analogical reasoning (Langlais and Patry, 2007).

Vilar et al. (2007) investigate a way to handle data sparsity in machine translation between closely related languages by translating between characters as opposed to words. The words in the parallel sentences are segmented into characters. Spaces between words are marked with a special character. The sequences of characters are then used to train a standard machine translation model and a language model with n-grams up to $n = 16$. They apply

their system to the translation between the related languages Spanish and Catalan, and find that a word based system outperforms their letter-based system. However, a combined system performs marginally better in terms of BLEU scores.

Tiedemann (2009) shows that combining character-based translation with phrase-based translation improves machine translation quality in terms of BLEU and NIST scores when translating between Swedish and Norwegian if the OOV-words are translated beforehand with the character-based model.

Nakov and Tiedemann (2012) investigate the use of character-level models in the translation between Macedonian and Bulgarian movie subtitles. Their aim is to translate between the resource poor language Macedonian to the related language Bulgarian, in order to use Bulgarian as a pivot in order to translate to other languages such as English. Their research shows that using character bigrams shows improvement over a word-based baseline.

It seems clear that character overlap can be used to improve translation quality in related languages. We plan to use character overlap in language transformation between two diachronic variants of a language.

5.1.2 *This study*

In this study we investigate the task of translating from Middle Dutch to Modern Dutch. Similarly to resource poor languages, one of the problems that are apparent is the small amount of parallel Middle Dutch - Modern Dutch data that is available. To combat the data sparseness we aim to use the character overlap that exists between the Middle Dutch words and their Modern Dutch counterparts. Examples of overlap in some of the words given in example 5.1 and 5.2 can be viewed in Table 5.1. We are interested in whether we can use this overlap to improve the performance of the translation model. We consider three approaches: (A) Perform normal PBMT without any preprocessing, (B) Apply a preprocessing step in which we pinpoint words and phrases that can be aligned based on character overlap and (C) perform machine translation not to words but to character bigrams in order to make use of the character overlap.

We will first discuss the PBMT baseline, followed by the PBMT + overlap system and the character bigram PBMT transliteration system in Section 5.2. We then describe the experiment with validation by human judges in Section 5.4, and its results in Section 5.5. We close with a discussion of our results in Section 5.6.

Middle Dutch	Modern Dutch
versamet	verzameld
was	was
niemen	niemand
die	de
das	das
claghene	klagen
over	over
Reynaerde	Reynaert
metten	met zijn
grijsen	grijze
baerde	baard

Table 5.1: Examples of character overlap in words from a fragment of 'Van den vos Reynaerde'

5.2 LANGUAGE TRANSFORMATION MODELS

5.2.1 *PBMT baseline*

For our baseline we use the Moses software to train a PBMT model (Koehn et al., 2007). In general, a statistical machine translation model normally finds a best translation $\tilde{E}$ of a text in language F for a text in language E by combining a translation model $P(F|E)$ with a language model $P(E)$:

$$\tilde{E} = \arg\max_{E \in E^*} P(F|E)P(E)$$

In phrase-based machine translation the sentence F is segmented into a sequence of I phrases during decoding. Each source phrase is then translated into a target phrase to form sentence E. Phrases may be reordered. This is described in more detail in Chapter 1.

The GIZA++ statistical alignment package is used to perform the word alignments, which are later combined into phrase alignments in the Moses pipeline (Och and Ney, 2003) to build the language transformation model. GIZA++ implements IBM Models 1 to 5 and an HMM word alignment

model to find statistically motivated alignments between words. We first tokenize our data. We then lowercase all data and use all sentences from the Modern Dutch part of the corpus to train an n-gram language model with the SRILM toolkit (Stolcke, 2002). Then we run the GIZA++ aligner using the training pairs of sentences in Middle Dutch and Modern Dutch. We execute GIZA++ with standard settings and we optimize using minimum error rate training with BLEU scores. The Moses decoder is used to generate the translations.

5.2.2 *PBMT with overlap-based alignment*

Before using the Moses pipeline we perform a preprocessing alignment step based on character overlap. Word and phrase pairs that exhibit a large amount of character overlap are added as a dictionary to the parallel corpus that GIZA++ is trained on. This helps to bias the alignment procedure towards aligning similar words and reduces alignment errors. To perform the preprocessing step we use the Needleman-Wunsch algorithm (Needleman and Wunsch, 1970). The Needleman-Wunsch algorithm is a dynamic programming algorithm that performs a global alignment on two sequences. Sequence alignment is a method to find commonalities in two (or more) sequences of some items or characters. One often used example is the comparison of sequences of DNA to find evolutionary differences and similarities. Sequence alignment is also used in linguistics, where it is applied to finding the longest common substring or the differences or similarities between strings.

The Needleman-Wunsch algorithm is a sequence alignment algorithm that optimizes a score function to find an optimal alignment of a pair of sequences. Each possible alignment is scored according to the score function, where the alignment giving the highest similarity score is the optimal alignment of a pair of sequences. If more than one alignment yields the highest score, there are multiple optimal solutions. The algorithm uses an iterative matrix to calculate the optimal solution. All possible pairs of characters containing one character from each sequence are plotted in a 2-dimensional matrix. Then, all possible alignments between those characters can be represented by pathways through the matrix. Insertions and deletions are allowed, but can be penalized by means of a gap penalty in the alignment.

The first step is to initialize the matrix and fill in the gap scores in the top row and leftmost column. In our case we heuristically set the values of the scores to +1 for matches, -2 for mismatches and -1 for gaps after evaluating on our development set. The initialization step can be seen in Table 5.2. After

		k	w	a	m
	0	-1	-2	-3	-4
q	-1				
u	-2				
a	-3				
m	-4				

Table 5.2: Initialization step of the Needleman-Wunsch algorithm, filling out the gap scores of the top row and left column.

		k	w	a	m
	0	-1	-2	-3	-4
q	-1	-2	-3	-4	-5
u	-2	-3	-4	-5	-6
a	-3	-4	-5	-3	-4
m	-4	-5	-6	-4	-2

Table 5.3: The filled out Needleman-Wunsch matrix. Each cell is filled with the maximum score of the three possible edit operations.

		k	w	a	m
	0	-1	-2	-3	-4
q	-1	**-2**	-3	-4	-5
u	-2	-3	**-4**	-5	-6
a	-3	-4	-5	**-3**	-4
m	-4	-5	-6	-4	**-2**

Table 5.4: The optimal alignment is a traceback through the cells with highest scores, starting at the lower right corner.

this step, we can label the cells in the matrix $C(i,j)$ where $i = 1, 2, ..., N$ and $j = 1, 2, ..., M$, the score of any cell $C(i,j)$ is then the maximum of:

$$\begin{aligned} q_{diag} &= C(i-1, j-1) + s(i,j) \\ q_{down} &= C(i-1, j) + g \\ q_{right} &= C(i, j-1) + g \end{aligned}$$

Here, $s(i,j)$ is the substitution score for letters i and j, and g is the gap penalty score. If i and j match, the substitution score is in fact the matching score. The table is filled this way recursively, filling each cell with the maximum score of the three possible options (diagonally, down and right). After this is done, an optimal path can be found by performing a traceback, starting in the lower right corner and ending in the upper left corner, by visiting cells horizontally, vertically or diagonally, but only those cells with the highest score. After this process we end up with an alignment, as can be observed in Table 5.4.

We use the Needleman-Wunsch algorithm to find an optimal alignment of the Middle Dutch - Modern Dutch sentence pairs. We regard each line as a sentence. In case of rhyming text, a frequent phenomenon in Middle Dutch text, lines are usually parts of sentences. We then consider each line a string, and we try to align as many characters and whitespaces to their equivalents in the parallel line. We split the aligned sentences in each position where two whitespaces align and we consider the resulting aligned words or phrases as alignments. For each aligned word- or phrase-pair we calculate the Jaccard coefficient and if that is equal or higher than 0.5 we add the aligned words or phrases to the training material. By using this method we already can find many-to-one and one-to-many alignments. In this way we help the GIZA++ alignment process by biasing it towards aligning words and phrases that show overlap. Table 5.5 illustrates this process for two lines.

5.2.3 *Character-based transliteration*

Another somewhat novel approach we propose for Language Transformation is character-based transliteration. To circumvent the problem of OOV-words and use the benefits of character overlap more directly in the MT system, we build a translation model based on character bigrams, similar to Nakov and Tiedemann (2012). Where they use this approach to translate between closely related languages, we use it to translate between diachronic variants of a language. Another difference is that we keep the segmentation into bigrams throughout the translation process. The sentences in the parallel corpus are broken into character bigrams, with a special character representing whitespaces. These bigrams are used to train the translation model and the language model. An example of the segmentation process is displayed in Table 5.6. We train an SRILM language model on the character bigrams and model sequences of up to 10 bigrams. We then run the standard Moses pipeline, using GIZA++ with standard settings to generate the phrase-table and we use the Moses decoder to decode the bigram sequences.

A number of sample entries are shown in Table 5.7. As a final step, we recombine the bigrams into words. The different sizes of the Phrase-table for the different approaches can be observed in Table 5.8.

Middle Dutch:	hine	hadde+	++te	claghene	over	Reynaerde	,
Modern Dutch:	di-e	++niet	kwam	klag-en-	over	Reynaer-t	,
Jaccard	0.4	0.14	0	0.63	1	0.70	1

Middle Dutch:	+den	fe++llen	met++ten	grijsen	baerde	.
Modern Dutch:	die+	deugniet	met zijn	grijze+	baard+	.
Jaccard	0.50	0.09	0.50	0.71	0.8	1

Table 5.5: Alignment of lines with Jaccard scores for the aligned phrases. A + indicates a gap introduced by the Needleman Wunsch alignment.

original	segmented
Hine	#H Hi in ne e#
hadde	#h ha ad dd de e#
te	#t te e#
claghene	#c cl la ag gh he en ne e#
over	#o ov ve er r#
Reynaerde	#R Re ey yn na ae er rd de e#
,	#, ,#

Table 5.6: Input and output of the character bigram segmenter

5.3 DATA SET

Our training data consists of various Middle Dutch literary works with their modern Dutch translation. A breakdown of the different works is in Table 5.9. All works are from the Digital Library of Dutch Literature[1]. "Middle Dutch" is a very broad definition. It encompasses all Dutch language variants spoken and written between 1150 and 1500 in the Netherlands and Flanders. Works stemming from different centuries, regions and writers can

1 http://www.dbnl.org

#d da at t#	en n# #d da aa ar
#d da at t#	et t# #s st
#d da at t#	et t# #s
#d da at t#	ie et t# #s st
#d da at t#	ie et t# #s
#d da at t#	la an n#
#d da at t#	le et t#
#d da at t#	n# #d da aa ar ro
#d da at t#	n# #d da aa ar
#d da at t#	n#
#d da at t#	rd da at t#
#d da at ts s#	#d da at t#
#d da at ts si i#	#h he eb bb be en
#d da at ts	#d da at t#
#d da at ts	#w wa at t#
#d da at tt tu	#w wa at t# #j

Table 5.7: Example entries from the character bigram Phrase-table, without scores.

system	lines
PBMT	20,092
PBMT + overlap	27,885
character bigram transliteration	93,594

Table 5.8: Phrase-table sizes of the different models

differ greatly in their orthography and spelling conventions. No variant of Dutch was considered standard or the norm; Middle Dutch can be considered a collection of related lects (regiolects, dialects). This only adds to the problem of data sparsity. Our test set consists of a selection of sentences from the Middle Dutch work Beatrijs. Beatrijs is a Maria legend, detailing the legend of a nun who deserts her convent for the love of a man. After many hardships she finds out Maria has acted in her role at the convent,

and she returns to the convent, without anyone knowing she had been away for years. It was written around 1374 by an anonymous author.

source text	lines	date of origin
Van den vos Reynaerde	3428	around 1260
The Reis van Sint Brandaan	2312	12th century
Gruuthuuse gedichten	224	around 1400
't Prieel van Trojen	104	13th century
Various poems	42	12th-14th centuries

Table 5.9: Middle Dutch works in the training set

5.4 EXPERIMENT

In order to evaluate the systems, we ran an experiment to collect human rankings of the output of the systems.

5.4.1 *Materials*

Because of the nature of our data, in which sentences often span multiple lines, it is hard to evaluate the output on the level of separate lines. Context is needed to properly evaluate such a line. We therefore choose to evaluate pairs of lines. We randomly choose a line, and check if it is part of a sensible sentence that can be understood without context. If that is the case, we select it to include in our test set. In this way we select 25 pairs of lines. We evaluate the translations produced by the three different systems for these sentences. Examples of the selected sentences and the generated corresponding output are displayed in Table 5.10.

5.4.2 *Participants*

The participants in this evaluation study were 22 volunteers. All participants were native speakers of Dutch, and participated through an online interface. All participants were adults, and 12 were male and 10 female. In addition to the 22 participants, one expert also performed the experiment, in order to be able to compare the judgements of the laymen and the expert.

Middle Dutch	Si seide: 'Ic vergheeft u dan. Ghi sijt mijn troest voer alle man
Modern Dutch	Ze zei: 'ik vergeef het je dan. Je bent voor mij de enige man
PBMT	Ze zei : ' Ik vergheeft u dan . Gij ze alles in mijn enige voor al man
PBMT + Overlap	Ze zei : ' Ik vergheeft u dan . dat ze mijn troest voor al man
Char. Bigram PBMT	Ze zeide : ' Ik vergeeft u dan . Gij zijt mijn troost voor alle man
Middle Dutch	Dat si daer snachts mochte bliven. 'Ic mocht u qualijc verdriven,'
Modern Dutch	omdat ze nu niet verder kon reizen. 'Ik kan u echt de deur niet wijzen,'
PBMT '	dat ze daar snachts kon bliven . Ik kon u qualijc verdriven , '
PBMT + Overlap	dat ze daar s nachts kon blijven . ' Ik kon u qualijc verdriven , '
Char. Bigram PBMT	dat zij daar snachts mocht blijven . ' Ik mocht u kwalijk verdrijven ,

Table 5.10: Example output

5.4.3 *Procedure*

The participants were asked to rank three different automatic literal translations of Middle Dutch text. For reference, they were also shown a modern (often not literal) translation of the text by Dutch author Willem Wilmink. The order of sentences was randomized, as well as the order of the output of the systems for each sentence. The participants were asked to consider the extent to which the sentences could be deciphered and understood in their ranking. The participants were asked to always provide a ranking and were not allowed to assign the same rank to multiple sentences. In this way,

each participant provided 25 rankings where each pair of sentences received a distinct rank. The pair with rank 1 is considered best and the pair with 3 is considered worst.

system	mean rank	95 % confidence interval
PBMT	2.44 (0.03)	2.38 - 2.51
PBMT + Overlap	2.00 (0.03)	1.94 - 2.06
character bigram PBMT	1.56 (0.03)	1.50 - 1.62

Table 5.11: Mean scores assigned by human subjects, with the standard error between brackets and the lower and upper bound of the 95 % confidence interval

5.5 RESULTS

5.5.1 *Human judgements*

In this section we report on results of an experiment with judges ranking the output of the systems. To test for significance of the difference in the ranking of the different systems we ran repeated measures analyses of variance with system (PBMT, PBMT + Overlap, character bigram MT) as the independent variable, and the ranking of the output as the dependent variable. Mauchly's test for sphericity was used to test for homogeneity of variance, but was not significant, so no corrections had to be applied. Planned pairwise comparisons were made with the Bonferroni method. The mean ranking can be found in Table 5.11 together with the standard deviation and 95 % confidence interval. We find that participants ranked the three systems differently, $F(2,42) = 135,604, p < .001, \eta_p^2 = .866$. All pairwise comparisons are significant at $p < .001$. The character bigram model receives the best mean rank (1.56), then the PBMT + Overlap system (2.00) and the standard PBMT system is ranked lowest (2.44). We ran a Friedman test on each of the 25 K-related samples, and found that for 13 sentences the ranking provided by the test subjects was equal to the mean ranking: the PBMT system ranked lowest, then the PBMT + Overlap system and the character bigram system scored highest for each of these cases at $p < .005$. These results are detailed in Table 5.12. When comparing the judgements of the participants with the judgements of an expert, we find a significant Pearson correlation of .65 ($p < .001$) between the judgements of the expert and the mean of

PBMT	PBMT + overlap	character bigram PBMT	X^2
2.05	2.59	1.36	16.636**
2.77	1.82	1.41	21.545**
2.50	1.27	2.23	18.273**
1.95	1.45	2.59	14.273**
2.18	2.36	1.45	10.182**
2.45	2.00	1.55	9.091*
2.91	1.77	1.32	29.545**
2.18	2.27	1.55	6.903*
2.14	2.00	1.86	0.818
2.27	1.73	2.00	3.273
2.68	1.68	1.64	15.364**
2.82	1.95	1.23	27.909**
2.68	2.09	1.23	23.545**
1.95	2.55	1.50	12.091**
2.77	1.86	1.36	22.455**
2.32	2.23	1.45	9.909**
2.86	1.91	1.23	29.727**
2.18	1.09	2.73	30.545**
2.05	2.09	1.86	0.636
2.73	2.18	1.09	30.545**
2.41	2.27	1.32	15.545**
2.68	2.18	1.14	27.364**
1.82	2.95	1.23	33.909**
2.73	1.95	1.32	21.909**
2.91	1.77	1.32	29.545**

Table 5.12: Results of the Friedman test on each of the 25 sentences. Results marked * are significant at $p < 0.05$ and results marked ** are significant at $p < 0.01$

the judgements of the participants. This indicates that the judgements of the laymen are indeed useful.

5.5.2 *Automatic judgements*

In order to attempt to measure the quality of the transformations made by the different systems automatically, we measured NIST scores by comparing the output of the systems to the reference translation. We do realize that the reference translation is in fact a literary interpretation and not a literal translation, making automatic assessment harder. Having said that, we still hope to find some effect by using these automatic measures. We only report NIST scores here, because BLEU turned out to be very uninformative. In many cases sentences would receive a BLEU score of 0. Mauchly's test for sphericity was used to test for homogeneity of variance for the NIST scores, and was not significant. We ran a repeated measures test with planned pairwise comparisons made with the Bonferroni method. We found that the NIST scores for the different systems differed significantly ($F(2,48) = 6.404, p < .005, \eta_p^2 = .211$). The average NIST scores with standard error and the lower and upper bound of the 95 % confidence interval can be seen in Table 5.13. The character bigram transliteration model scores highest with 2.43, followed by the PBMT + Overlap model with a score of 2.30 and finally the MT model scores lowest with a NIST score of 1.95. We find that the scores for the PBMT model differ significantly from the PBMT + Overlap model ($p < .01$) and the character bigram PBMT model ($p < .05$), but the scores for the PBMT + Overlap and the character bigram PBMT model do not differ significantly. When we compare the automatic scores to the human assigned ranks we find no significant Pearson correlation.

system	mean NIST score	95 % confidence interval
PBMT	1.96 (0.18)	1.58 - 2.33
PBMT + overlap	2.30 (0.21)	1.87 - 2.72
character bigram PBMT	2.43 (0.20)	2.01 - 2.84

Table 5.13: Mean NIST scores, with the standard error between brackets and the lower and upper bound of the 95 % confidence interval

5.6 CONCLUSION

In this chapter we have described two modifications of the standard PBMT framework to improve the transformation of Middle Dutch to Modern Dutch by using character overlap in the two variants. We described one approach that helps the alignment process by adding words that exhibit a certain amount of character overlap to the parallel data. We also described another approach that translates sequences of character bigrams instead of words. Reviewing the results we conclude that the use of character overlap between diachronic variants of a language is beneficial in the translation process. More specifically, the model that uses character bigrams in translation instead of words is ranked best. Also ranked significantly better than a standard phrase-based machine translation approach is the approach using the Needleman-Wunsch algorithm to align sentences and identify words or phrases that exhibit a significant amount of character overlap to help the GIZA++ statistical alignment process towards aligning the correct words and phrases. We have seen that one issue that is encountered when considering the task of language transformation from Middle Dutch to Modern Dutch is data sparseness. The amount of lines used to train on amounts to a few thousand, and not millions as is more common in SMT. It is therefore crucial to use the inherent character overlap in this task to compensate for the lack of data and to make more informed decisions. The character bigram approach is able to generate a translation for out of vocabulary words, which is also a solution to the data sparseness problem. One area where the character bigram model often fails is in translating Middle Dutch words into Modern Dutch words that are significantly different. One example can be seen in Table 5.10, where 'mocht' is translated by the PBMT and PBMT + Overlap systems to 'kon' and left the same by the character bigram transliteration model. This is probably due to the fact that 'mocht' still exists in Dutch, but is not as common as 'kon' (meaning 'could'). Another issue to consider is the fact that the character bigram model learns character mappings that are occur troughout the language. One example is the disappearing silent 'h' after a 'g'. This often leads to transliterated words of which the spelling is only partially correct. Apparently the human judges rate these 'half words' higher than completely wrong words, but automatic measures such as NIST are insensitive to this.

We have also reported the NIST scores for the output of the standard PBMT approach and the two proposed variants. We see that the NIST scores show a similar pattern as the human judgements: the PBMT + Overlap and character bigram PBMT systems both achieve significantly higher NIST scores than the normal PBMT system. However, the PBMT + Overlap and

character bigram PBMT models do not differ significantly in scores. Interesting is that we find no significant correlation between human and automatic judgements, leading us to believe that automatic judgements are not viable in this particular scenario. This is perhaps due to the fact that the reference translations the automatic measures rely on are in this case not literal translations but rather literary interpretations of the source text in modern Dutch. The fact that both versions are written in rhyme only worsens this problem, as the author of the Modern Dutch version is often very creative.

We think techniques such as the ones described here can be of great benefit to laymen wishing to investigate works that are not written in contemporary language, resulting in improved access to these older works. Our character bigram transliteration model may also play some role as a computational model in the study of the evolution of orthography in language variants, as it often will generate words that are strictly speaking not correct, but do resemble Modern Dutch in some way.

6

GENERAL DISCUSSION AND CONCLUSION

In this thesis we studied text-to-text generation. We argued text-to-text generation can be seen as a monolingual machine translation problem, and we applied monolingual phrase-based machine translation (PBMT) to several text-to-text generation tasks, namely: sentential paraphrase generation, sentence simplification, sentence compression, and language transformation.

6.1 STUDY 1: PARAPHRASE GENERATION

In Chapter 2 we investigated an approach to collect a parallel sentential paraphrase corpus of a size that is comparable with bilingual parallel corpora used for statistical machine translation. We described a method that used a news aggregator site (Google News) to collect headlines and aligned these headlines based on word overlap. Using this approach a considerable number of paraphrases can be aligned for which headlines are available on the news aggregator service. By using this new data collection approach, paraphrase generation approaches that are based on parallel monolingual corpora become more viable, because they can be trained on considerably larger amounts of data. To generate the sentential paraphrases we used a PBMT approach with re-ranking of the ten best translations based on dissimilarity with the source sentence measured by word Levenshtein distance (PBMT-R). This approach was used for paraphrase generation in Dutch and English. We evaluated the output of the paraphrase system for both Dutch

and English by using human judges. These judges evaluated the output on the dimensions of fluency (the extent to which the sentence is well formed) and adequacy (the extent to which the information contained in the source sentence is retained in the target sentence). Edit distance was evaluated automatically by using Levenshtein distance on the sentence level, with each token being an atomic unit upon which edit operations can be performed. The judges preferred the output of the PBMT-R system to an informed word substitution baseline.

6.2 STUDY 2: SENTENCE SIMPLIFICATION

In Chapter 3 we demonstrated that the PBMT-R model can directly be ported to the task of sentence simplification. We used the PWKP (Zhu et al., 2010) data set, which contains paired sentences from normal English Wikipedia and Simple English Wikipedia, to train our system on. We compared our system to two state of the art sentence simplification systems trained on similar data (a tree-based translation system by Zhu et al. (2010) and a synchronous tree grammar mode by Woodsend and Lapata (2011) and an informed word substitution baseline. Human judges evaluated the output of the systems on three dimensions: fluency, adequacy and simplicity. While achieving a similar level of simplicity as the two state of the art systems, the PBMT-R model scored significantly better in terms of fluency and adequacy. The PBMT-R model also outperformed the baseline on the dimensions of simplicity and adequacy. This indicates that while it is conservative in its edits and does not perform sentence splitting, the PBMT-R model can be successfully applied to the task of sentence simplification.

6.3 STUDY 3: SENTENCE COMPRESSION

In Chapter 4 we investigated the task of sentence compression. We compared extractive and abstractive sentence compression. To build an extractive model we used a memory-based deletion approach and trained this model on an extractive corpus of sentences from news reports paired with manually created extractive compressions. We showed that our extractive model can easily be tuned to perform more compression or less compression by varying the k nearest neighbors in the deletion classifier. We combined this model with the PBMT-R model trained on the PWKP dataset in order to create an abstractive model. We tested the systems on the abstractive sentence compression corpus by Cohn and Lapata (2008). Human judges evaluated the output of the systems on two dimensions: fluency and adequacy.

Compression was measured by using character compression rate, and was kept the same for both systems. We showed that our systems compress to a higher degree than state of the art systems, such as the extractive and abstractive systems by Cohn and Lapata (2008), but not as much as humans. Human judges preferred the extractive compression system over the abstractive compression system on both fluency and adequacy.

6.4 STUDY 4: LANGUAGE TRANSFORMATION

In Chapter 5 we investigated the use of the PBMT in language transformation. We defined language transformation as translating between diachronically distinct language variants. Our focus was on the transformation from Middle Dutch to Modern Dutch. We described three systems to tackle this task: a standard PBMT approach, a PBMT approach augmented with overlap-based alignment, and a character bigram PBMT transliteration approach. Human judges were asked to rank the output of the systems. We found that the three approaches were ranked significantly differently, with the transliteration PBMT approach being ranked first and the PBMT approach augmented with overlap-based alignment being ranked second.

6.5 ANSWERS TO THE RESEARCH QUESTIONS

Taking all this together we can now answer the research questions posed in Chapter 1.

1. How can a statistical machine translation model be applied to a collection of monolingual text-to-text generation tasks?

We have shown that phrase-based machine translation combined with a re-ranking heuristic (PBMT-R) can be applied to a collection of text-to-text generation tasks by applying relatively small modifications to the model. The re-ranking heuristic that we use is based on Levenshtein distance applied to the word level. This makes sure the model generates output that differs from its input. For paraphrase generation, the PBMT-R approach outperforms an informed word-substitution baseline. In the case of sentence simplification the PBMT-R model outperforms two state of the art systems on adequacy and fluency, and performs similarly on simplification. We have demonstrated that for language transformation various modifications can be made to the PBMT model that exploit character overlap and improve the performance of PBMT for this task. We have also demonstrated that character bigram transliteration can easily be implemented in the PBMT model, which

has the advantage that it exploits character overlap between diachronic variants and that it can translate words that it has not encountered before in its training data.

For sentence compression, however, the PBMT-R scores are lower than the scores of the extractive approach. This can be attributed to several reasons. One possible reason is that the simplification data is not well suited for compression. Although on average the sentences are shorter in Simple Wikipedia compared to normal Wikipedia, the aim of Simple Wikipedia is not to provide compressions, but rather to provide sentences that are easier to understand. Another potential reason is that the extractive approach is a very straightforward one that is relatively easy to achieve and yet produces sentence compression quite successfully. A third reason is the nature of the task: in machine translation, it is generally bad to delete words from the sentence, but in sentence compression we do want to delete words if they do not convey the most important information of the sentence.

In general, as long as the task can be cast as a sentence-to-sentence generation task it is possible to successfully use monolingual machine translation. There are some limitations to the model. It is not suitable for operations such as sentence splitting and deletion. This problem can be solved by combining the PBMT-R model with other modules that perform these operations.

2. How can good parallel monolingual corpora be created?

For all four tasks we used parallel monolingual data, although the size of the data collections differed from task to task. We have introduced a new way of collecting large sentential paraphrase corpora from monolingual data for English and Dutch. We showed how to collect news headlines from news aggregator sites such as Google News. Headlines are a good source for paraphrases because different news outlets have different ways of formulating their headlines, yet they still describe the same event. This means that for each event we get a collection of headlines which is a potential rich resource of sentential paraphrases. Using overlap based measures we can then extract paraphrase pairs. One advantage is that in this way a great amount of data can be collected. Another advantage is that news aggregators are available in various languages, making it easy to use this approach for various languages. A third advantage of these data is the nature of the data; the headlines we collect are actual paraphrases produced by news editors. This means that they are not a byproduct of another process, which is the case in the pivot approach. This also means that actual world knowledge is captured in the data. It can be argued that headlines use a particular form of language that is generally shorter than normal English, but we believe the

paraphrase patterns that are learned from these data can also be applied to other domains. We see our method as a valuable contribution to paraphrase research in multiple languages and the first to collect sentential paraphrases on this scale.

For simplification, we used data extracted from the normal English Wikipedia and the Simple English Wikipedia. Sentence pairs extracted from articles from Wikipedia and Simple Wikipedia can be used to model simplification, because the sentences in Simple Wikipedia are in essence simplifications of the sentences in Wikipedia. However, the mapping of sentences is not always straightforward, as sometimes sentences are split or fused, or sometimes relevant information is dropped from the sentence. We used an existing dataset by Zhu et al. (2010), the PWKP dataset, which has been used in earlier research on sentence simplification.

We used the extractive broadcast corpus by Clarke and Lapata (2008) to train our memory-based extractive sentence compressor. We again used the PWKP dataset to train the abstractive part of our sentence compression system, which consists of the PBMT-R model. This abstractive approach was unable to outperform the extractive model, when tested on the abstractive sentence compression developed corpus by Cohn and Lapata (2008).

For language transformation we could only use the parallel data that is available for works that have been manually translated into another diachronic variant. Typically this means not much data is available for language transformation. We have shown, however, that we can use characteristics of these data to overcome the relative sparseness of these data. We leverage the fact that the two variants of the language share some characteristics and indeed share a lot of characters in the words that are used. This can be attributed to the fact that the two languages use diachronic variants of the same words. For our language transformation study we used a collection of parallel texts for Middle Dutch from the Digital Library of Dutch Literature.

In general the success of collecting monolingual corpora depends on the task the corpora are used for. For paraphrasing, large amounts of data can be found, as paraphrases are abundant in news headlines and can be found relatively easily. In order to construct simplification corpora, the task of constructing corpora is changed into finding simplifications of sentences. This task is a lot harder, because less sources are available. This also holds true for compression and transformation.

3. To what extent can text-to-text generation be evaluated automatically?

In order to evaluate the various text-to-text generation applications, we used human judges in addition to automatic measures. Using automatic

measures is an accepted procedure in machine translation, as these have been demonstrated to correlate to a high degree with human judgements by Papineni et al. (2002) . In monolingual text-to-text generation human judgements are used typically, because in general there are no automatic metrics which have been proven to correlate highly with human judgements for these tasks. In all of our studies we observed that the machine translation evaluation metrics such as BLEU and NIST and the ROUGE summarization metric showed the same tendencies as the human judgements. In order to investigate the legitimacy of these results we calculated correlations between human judgements and automatic metrics.

For the paraphrase generation experiment we found significant correlations between the automatic metrics (BLEU, NIST) and the human judgements (adequacy, fluency). These correlations were however in the low regions (below 0.15). Furthermore, we found that when we split the results according to edit distance with tokens as atomic units, we can perform a more meaningful automatic evaluation. As the edit distance increases, automatic scores for systems that are judged to be better tend to show a less steep decline than those of the lower rated systems. For the sentence simplification task we measured correlations between BLEU and human judgements (adequacy, fluency, simplicity) and found significant low to medium correlations between BLEU and fluency and simplicity. We also used the Flesch-Kincaid grade level readability metric and demonstrated that it is unsuitable for the evaluation of text-to-text generation. This can be explained by the fact that this metric only factors in the number of words per sentence and the number of syllables per word. It does not say anything about the content or the grammaticality of the sentence. For sentence compression and language transformation we found no significant correlation between automatic metrics and human judgements.

We conclude that although scores assigned by automatic metrics may be indicative of the quality of text-to-text generation systems, they should be used with caution, as they are not well suited to handle the large amount of variety that can occur in the output of monolingual text-to-text generation systems. One way to deal with this is for automatic metrics to use a large amount of reference translations (in the monolingual case often paraphrases). This means the metric can compare the output to multiple references, and is therefore more likely to match varying output to a reference. We were able to use multiple references in the paraphrase study, but not in the other studies. We conclude that automatic evaluation can be used to gain some insight in the performance of text-to-text generation systems (for example in system development), but to get the full picture manual evaluation is still needed.

6.6 FUTURE WORK

One important direction for future research is improving the PBMT-R model. We have found that in general it is quite conservative in the edits it performs and it is limited to single sentences. We have approached text-to-text generation as a sentence-to-sentence task, just as machine translation. However, text modification or translation is in general more than a strict sentence-to-sentence task. As long as the task can be brought down to multiple sentence-to-sentence generations, it is possible to use our approach for these tasks. Combining the PBMT-R system with systems that perform typical text-to-text operations such as sentence splitting and more radical syntax alterations would be interesting to implement. Another thing that our model does not take into account is the context of a sentence. Context may be needed to provide an accurate translation of a sentence, or in our case for instance a correct paraphrase. This is one of the problems that is generally not addressed in machine translation, and we see this as an interesting direction for future research. Another interesting direction is the application of our models to other domains. Applying for instance our paraphrase model on other domains than headlines would give us more insight in the applicability of the model in those domains.

Another direction of future research is the collection of more diverse corpora. We have demonstrated how to compile a corpus of paraphrasing headlines, but as we mentioned earlier many more domains remain. For simplification and compression sources such as Wikipedia edit histories or pairs of scientific articles and abstracts might be used. In the process of editing Wikipedia many editors add, rephrase or remove information. Edits that shorten sentences can be extracted and used to create such a corpus. It would be very valuable to collect these corpora in order to improve performance on these tasks. This also holds for the language transformation task, where we only used small amounts of data. Obtaining more data could greatly benefit this task.

The automatic evaluation of monolingual text-to-text generation remains an unsolved issue as well. We have demonstrated that breaking down the evaluation to edit distance for paraphrasing or compression rate for sentence compression can aid the evaluation process, but in order to have fully automatic evaluation the issue of coverage arises. For paraphrase generation, the goal is to create a different variant of a sentence. Any successful automatic metric should be able to handle this variety of possible generated phrases. Automatic metrics that correlate well with human judgements could greatly help research in text-to-text generation. These metrics should take into account fluency, adequacy as well as additional constraints such

as degree of differentness for paraphrase generation or degree of simplification for sentence simplification. Additional manual evaluation could also be valuable for text-to-text generation research. Evaluation of simplification can for example be done in an application for low literacy readers or people with reading disabilities, in order to investigate how much they benefit from using such an application.

6.7 CONCLUSION

In sum, we have demonstrated that a machine translation framework can be used in the context of monolingual text-to-text generation tasks. We have shown how a phrase-based machine translation system can be modified to tackle various monolingual text-to-text generation tasks. We believe this approach can be easily adapted to various text-to-text generation tasks in multiple languages. We have also detailed an approach to collect sentential paraphrases on a large scale. These contributions combined result in a method to quickly build robust yet elegant models that can be used for various monolingual text-to-text generation tasks on the sentence level. Apart from these tasks themselves, they can also serve as modules in more sophisticated natural language processing pipelines, such as summarization on the document level, or automatic subtitling, as long as the problem can be brought down to a sentence-to-sentence generation task.

BIBLIOGRAPHY

Aha, D. W., Kibler, D., and Albert, M. K. (1991). Instance-based learning algorithms. *Mach. Learn.*, 6(1):37–66.

Androutsopoulos, I. and Malakasiotis, P. (2010). A survey of paraphrasing and textual entailment methods. *J. Artif. Int. Res.*, 38:135–187.

Anick, P. G. and Tipirneni, S. (1999). The paraphrase search assistant: terminological feedback for iterative information seeking. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '99, pages 153–159, New York, NY, USA. ACM.

Bannard, C. and Burch, C. C. (2005). Paraphrasing with bilingual parallel corpora. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 597–604, Morristown, NJ, USA. Association for Computational Linguistics.

Baron, A. and Rayson, P. (2008). VARD 2: A tool for dealing with spelling variation in historical corpora. In *Proceedings of the Postgraduate Conference in Corpus Linguistics*, Birmingham, UK. Aston University.

Barzilay, R. and Elhadad, N. (2003). Sentence alignment for monolingual comparable corpora. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 25–32, Morristown, NJ, USA. Association for Computational Linguistics.

Barzilay, R. and Lee, L. (2003). Learning to paraphrase: an unsupervised approach using multiple-sequence alignment. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 16–23, Morristown, NJ, USA. Association for Computational Linguistics.

Barzilay, R., McKeown, K., and Elhaded, M. (1999). Information fusion in the context of multi-document summarization. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL-99)*, Maryland.

Barzilay, R. and McKeown, K. R. (2001). Extracting paraphrases from a parallel corpus. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, ACL '01, pages 50–57, Stroudsburg, PA, USA. Association for Computational Linguistics.

Bateman, J. (1997). Enabling technology for multilingual natural language generation: the KPML development environment. *Natural Language Engineering*, 3(1):15–55.

Belz, A. (2008). Automatic generation of weather forecast texts using comprehensive probabilistic generation-space models. *Natural Language Engineering*, 14(4):431.

Bhagat, R. and Ravichandran, D. (2008). Large scale acquisition of paraphrases for learning surface patterns. In *Proceedings of ACL/HLT*.

Brown, P., Cocke, J., Della Pietra, S., Della Pietra, V., Jelinek, F., Mercer, R., and Poossin, P. (1990). A statistical approach to language translation. *Computational Linguistics*, 16(2).

Brown, P., Della Pietra, S., Della Pietra, V., and Mercer, R. (1993). The mathematics of machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Burger, J. and Ferro, L. (2005). Generating an entailment corpus from news headlines. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, pages 49–54. Association for Computational Linguistics.

Callison-Burch, C. (2008). Syntactic constraints on paraphrases extracted from parallel corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 196–205, Stroudsburg, PA, USA. Association for Computational Linguistics.

Callison-Burch, C., Cohn, T., and Lapata, M. (2008). Parametric: an automatic evaluation metric for paraphrasing. In *COLING '08: Proceedings of the 22nd International Conference on Computational Linguistics*, Morristown, NJ, USA. Association for Computational Linguistics.

Callison-Burch, C., Koehn, P., and Osborne, M. (2006a). 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, Morristown, NJ, USA. Association for Computational Linguistics.

Callison-Burch, C., Osborne, M., and Koehn, P. (2006b). Re-evaluating the role of BLEU in machine translation research. In *In EACL*, pages 249–256.

Canning, Y., Tait, J., Archibald, J., and Crawley, R. (2000). Cohesive regeneration of syntactically simplified newspaper text. In *Proceedings of ROMAND 2000*, Lausanne.

Carroll, J., Minnen, G., Canning, Y., Devlin, S., and Tait, J. (1998). Practical simplification of English newspaper text to assist aphasic readers. In *AAAI-98 Workshop on Integrating Artificial Intelligence and Assistive Technology*, Madison, Wisconsin.

Carroll, J., Minnen, G., Pearce, D., Canning, Y., Devlin, S., and Tait, J. (1999). Simplifying text for language-impaired readers. In *Proceedings of EACL'99*, Bergen. ACL.

Chandrasekar, R., Doran, C., and Srinivas, B. (1996). Motivations and methods for text simplification. In *Proceedings of the Sixteenth International Conference on Computational Linguistics (COLING'96)*, pages 1041–1044.

Chandrasekar, R. and Srinivas, B. (1997). Automatic rules for text simplification. *Knowledge-Based Systems*, 10:183–190.

Chen, D. L. and Dolan, W. B. (2011). Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 190–200, Stroudsburg, PA, USA. Association for Computational Linguistics.

Chiang, D., Lopez, A., Madnani, N., Monz, C., Resnik, P., and Subotin, M. (2005). The Hiero machine translation system: extensions, evaluation, and analysis. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 779–786, Stroudsburg, PA, USA. Association for Computational Linguistics.

Clarke, J. and Lapata, M. (2008). Global inference for sentence compression an integer linear programming approach. *Journal of Artificial Intelligence Research*, 31:399–429.

Cohn, T., Callison-Burch, C., and Lapata, M. (2008). Constructing corpora for development and evaluation of paraphrase systems. *Computational Linguistics*, 34(4):597–614.

Cohn, T. and Lapata, M. (2007). Large margin synchronous generation and its application to sentence compression. In *Proceedings of EMNLP-CoLing*.

Cohn, T. and Lapata, M. (2008). Sentence compression beyond word deletion. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 137–144. Association for Computational Linguistics.

Corston-Oliver, S. (2001). Text compaction for display on very small screens. In *Proceedings of the Workshop on Automatic Summarization (WAS 2001)*, pages 89–98, Pittsburgh, PA, USA.

Coster, W. and Kauchak, D. (2011). Learning to simplify sentences using Wikipedia. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, pages 1–9, Portland, Oregon. Association for Computational Linguistics.

Daelemans, W. (1999). Introduction to the special issue on memory-based language processing. *Journal of Experimental & Theoretical Artificial Intelligence*, 11(3):287–296.

Daelemans, W., Hothker, A., and Tjong Kim Sang, E. (2004). Automatic sentence simplification for subtitling in Dutch and English. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, pages 1045–1048.

Daelemans, W., Zavrel, J., Berck, P., and Gillis, S. (1996). MBT: A memory-based part of speech tagger-generator. In *Proc. of Fourth Workshop on Very Large Corpora*, pages 14–27. ACL SIGDAT.

Daelemans, W., Zavrel, J., Van der Sloot, K., and Van den Bosch, A. (2009). TiMBL: Tilburg Memory Based Learner, version 6.2, reference manual. Technical Report ILK 09-01, Induction of Linguistic Knowledge, Tilburg University.

Daume, H. and Marcu, D. (2005). Induction of word and phrase alignments for automatic document summarization. *Computational Linguistics*, 31(4):505–530.

De Gispert, A. and Marino, J. B. (2006). Linguistic knowledge in statistical phrase-based word alignment. *Natural Language Engineering*, 12(1):91–108.

Deléger, L., Merkel, M., and Zweigenbaum, P. (2009). Translating medical terminologies through word alignment in parallel text corpora. *J. of Biomedical Informatics*, 42:692–701.

Doddington, G. (2002). Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research*, HLT '02, pages 138–145, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Dolan, B., Quirk, C., and Brockett, C. (2004). Unsupervised construction of large paraphrase corpora: exploiting massively parallel news sources. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, page 350, Morristown, NJ, USA. Association for Computational Linguistics.

Dras, M. (1997). Representing paraphrases using synchronous tags. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*, pages 516–518. Association for Computational Linguistics.

Elhadad, N. and Sutaria, K. (2007). Mining a lexicon of technical terms and lay equivalents. In *Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing*, BioNLP '07, pages 49–56, Stroudsburg, PA, USA. Association for Computational Linguistics.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*. The MIT Press.

Filippova, K. and Strube, M. (2008). Sentence fusion via dependency graph compression. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 177–185, Honolulu, Hawaii. Association for Computational Linguistics.

Finch, A., Dixon, P., and Sumita, E. (2012). Rescoring a phrase-based machine transliteration system with recurrent neural network language models. In *Proceedings of the 4th Named Entity Workshop (NEWS) 2012*, pages 47–51, Jeju, Korea. Association for Computational Linguistics.

Gale, W. A. and Church, K. W. (1993). A program for aligning sentences in bilingual corpora. *Computational Linguistics*, 19(1):75–102.

Galley, M. and McKeown, K. (2007). Lexicalized Markov grammars for sentence compression. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 180–187, Rochester, New York. Association for Computational Linguistics.

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*, EMNLP '11, pages 1168–1179, Stroudsburg, PA, USA. Association for Computational Linguistics.

Goldberg, E., Driedger, N., and Kittredge, R. (1994). Using natural-language processing to produce weather forecasts. *IEEE Expert*, 9(2):45–53.

Hajič, J., Hric, J., and Kuboň, V. (2000). Machine translation of very close languages. In *Proceedings of the sixth conference on Applied natural language processing*, pages 7–12. Association for Computational Linguistics.

Harris, Z. (1954). Distributional structure. *Word*, 10(23):46–162.

Heilman, M. and Smith, N. A. (2010). Extracting simplified statements for factual question generation. In *The Third Workshop on Question Generation*.

Hendrickx, I., Daelemans, W., Marsi, E., and Krahmer, E. (2009). Reducing redundancy in multi-document summarization using lexical semantic similarity. In Belz, A., Evans, R., and Varges, S., editors, *Proceedings of the ACL-IJCNLP 2009 Workshop: Language Generation and Summarisation (UCNLG+Sum)*, pages 63–66, Singapore. Association for Computational Linguistics.

Inui, K., Fujita, A., Takahashi, T., Iida, R., and Iwakura, T. (2003). Text simplification for reading assistance: A project note. In *Proceedings of the Second International Workshop on Paraphrasing*, pages 9–16, Sapporo, Japan. Association for Computational Linguistics.

Jing, H. and McKeown, K. (2000). Cut and paste based text summarization. In *Proceedings of the 1st Conference of the North American Chapter of the Association for Computational Linguistics*, pages 178–185, San Francisco, CA, USA.

Kauchak, D. and Barzilay, R. (2006). Paraphrasing for automatic evaluation. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 455–462, New York City, USA. Association for Computational Linguistics.

Kestemont, M., Daelemans, W., and De Pauw, G. (2010). Weigh your words—memory-based lemmatization for Middle Dutch. *Literary and Linguistic Computing*, 25(3):287–301.

Klein, D. and Manning, C. D. (2003). Accurate unlexicalized parsing. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 423–430, Morristown, NJ, USA. Association for Computational Linguistics.

Knight, K. and Marcu, D. (2000). Statistics-based summarization – step one: Sentence compression. In *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI)*, pages 703 – 710, Austin, Texas, USA.

Knight, K. and Marcu, D. (2002). Summarization beyond sentence extraction: a probabilistic approach to sentence compression. *Artif. Intell.*, 139(1):91–107.

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5.

Koehn, P., Hoang, H., Birch, A., Burch, C. 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 *ACL*. The Association for Computer Linguistics.

Koehn, P. and Knight, K. (2003). Feature-rich statistical translation of noun phrases. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 311–318, Morristown, NJ, USA. Association for Computational Linguistics.

Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 48–54. Association for Computational Linguistics.

Kondrak, G., Marcu, D., and Knight, K. (2003). Cognates can improve statistical translation models. In *HLT-NAACL*.

Kraaij, W. and Pohlmann, R. (1994). Porter's stemming algorithm for Dutch. In *Informatiewetenschap 1994: Wetenschappelijke bijdragen aan de derde STINFON Conferentie*, pages 167–180.

Krahmer, E. and Theune, M., editors (2010). *Empirical Methods in Natural Language Generation*. Lecture Notes in Computer Science 5790. Springer Verlag.

Langlais, P. and Patry, A. (2007). Translating unknown words by analogical learning. In *EMNLP-CoNLL*, pages 877–886. ACL.

Levenshtein, V. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710.

Li, Z., Callison-Burch, C., Dyer, C., Ganitkevitch, J., Khudanpur, S., Schwartz, L., Thornton, W. N. G., Weese, J., and Zaidan, O. F. (2009). Joshua: an open source toolkit for parsing-based machine translation. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 135–139, Stroudsburg, PA, USA. Association for Computational Linguistics.

Lin, C.-Y. (2003). Improving summarization performance by sentence compression - A pilot study. In *Proceedings of the Sixth International Workshop on Information Retrieval with Asian Languages*, volume 2003, pages 1–9.

Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Proc. ACL workshop on Text Summarization Branches Out*, page 10.

Lin, D. and Pantel, P. (2001a). Dirt: Discovery of inference rules from text. In *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 323–328, New York, NY, USA. ACM.

Lin, D. and Pantel, P. (2001b). Discovery of inference rules for question answering. *Natural Language Engineering*, 7(4):343–360.

Liu, C., Dahlmeier, D., and Ng, H. T. (2010). Pem: a paraphrase evaluation metric exploiting parallel texts. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 923–932, Stroudsburg, PA, USA. Association for Computational Linguistics.

Liu, F. and Liu, Y. (2009). From extractive to abstractive meeting summaries: can it be done by sentence compression? In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, ACLShort '09, pages 261–264, Stroudsburg, PA, USA. Association for Computational Linguistics.

Madnani, N., Ayan, N. F., Resnik, P., and Dorr, B. J. (2007). Using paraphrases for parameter tuning in statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, StatMT '07, pages 120–127, Stroudsburg, PA, USA. Association for Computational Linguistics.

Madnani, N. and Dorr, B. J. (2010). Generating phrasal and sentential paraphrases: A survey of data-driven methods. *Computational Linguistics*, 36(3):341–387.

Mani, I. (2001). *Automatic Summarization*. John Benjamins Publishers.

Marsi, E. and Krahmer, E. (2005). Explorations in sentence fusion. In *Proceedings of the 10th European Workshop on Natural Language Generation*, Aberdeen, GB.

Marsi, E. and Krahmer, E. (2007). Annotating a parallel monolingual treebank with semantic similarity relations. In *Proceedings of the 6th International Workshop on Treebanks and Linguistic Theories*, pages 85–96, Bergen, Norway.

Marsi, E., Krahmer, E., Hendrickx, I., and Daelemans, W. (2010). On the limits of sentence compression by deletion. In Krahmer, E. and Theune, M., editors, *Empirical methods in natural language generation*, pages 45–66. Springer-Verlag, Berlin, Heidelberg.

Marton, Y., Callison-Burch, C., and Resnik, P. (2009). Improved statistical machine translation using monolingually-derived paraphrases. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 381–390, Singapore. Association for Computational Linguistics.

McDonald, D. and Pustejovsky, J. (1985). Description-directed natural language generation. In *Proceedings of the 9th IJCAI*, pages 799–805.

McDonald, R. (2006). Discriminative sentence compression with soft syntactic evidence. In *Proceedings of EACL*.

McKeown, K. R. (1979). Paraphrasing using given and new information in a question-answer system. In *Proceedings of ACL*.

Nakov, P. and Tiedemann, J. (2012). Combining word-level and character-level models for machine translation between closely-related languages. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 301–305, Jeju Island, Korea. Association for Computational Linguistics.

Napoles, C., Callison-Burch, C., and Durme, B. V. (2011). Evaluating sentence compression: Pitfalls and suggested remedies. In *Workshop on Monolingual Text-To-Text Generation*.

Needleman, S. B. and Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453.

Nelken, R. and Shieber, S. M. (2006). Towards robust context-sensitive sentence alignment for monolingual corpora. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-06)*, Trento, Italy.

Neto, J., Freitas, A., and Kaestner, C. (2002). Automatic text summarization using a machine learning approach. *Advances in Artificial Intelligence*, pages 205–215.

Och, F. J. and Ney, H. (2000a). Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 440–447.

Och, F. J. and Ney, H. (2000b). Statistical machine translation. In *EAMT Workshop*, pages 39–46, Ljubljana, Slovenia.

Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Och, F. J., Tillmann, C., and Ney, H. (1999). Improved alignment models for Statistical Machine Translation. In *Proceedings of the Joint Workshop on Empirical Methods in NLP and Very Large Corpora*, pages 20–28, Maryland, USA.

Ogden, C. and Graham, E. (1935). *Basic English*. Anonymus-kiadás.

Pado, S., Galley, M., Jurafsky, D., and Manning, C. (2009). Robust machine translation evaluation with entailment features. In *Proceedings of ACL-IJCNLP*, pages 297–305.

Pang, Knight, and Marcu (2003). Syntax-based alignment of multiple translations: Extracting paraphrases and generating new sentences. In *HLT-NAACL*.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). BLEU: a method for automatic evaluation of machine translation. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318, Morristown, NJ, USA. Association for Computational Linguistics.

Paris, C., Swartout, W., and Mann, W. (1991). *Natural language generation in artificial intelligence and computational linguistics*. Kluwer Academic.

Pedersen, T. and Kulkarni, A. (2006). Automatic cluster stopping with criterion functions and the gap statistic. In *Proceedings of the 2006 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 276–279, Morristown, NJ, USA. Association for Computational Linguistics.

Popovic, M. and Ney, H. (2004). Towards the use of word stems and suffixes for statistical machine translation. In *LREC*. European Language Resources Association.

Quirk, C., Brockett, C., and Dolan, W. (2004). Monolingual machine translation for paraphrase generation. In Lin, D. and Wu, D., editors, *Proceedings of EMNLP 2004*, pages 142–149, Barcelona, Spain. Association for Computational Linguistics.

Ratnaparkhi, A. (2000). Trainable methods for surface natural language generation. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pages 194–201. Association for Computational Linguistics.

Ravichandran, D. and Hovy, E. (2002). Learning sufrace text patterns for a question answering system. In *Proceedings of ACL*.

Reiter, E. and Dale, R. (1997). Building applied natural language generation systems. *Natural Language Engineering*, 3(1):57–87.

Reiter, E. and Dale, R. (2000). *Building Natural Language Generation Systems*. Cambridge University Press, Cambridge.

Riezler, S., Vasserman, A., Tsochantaridis, I., Mittal, V. O., and Liu, Y. (2007). Statistical machine translation for query expansion in answer retrieval. In *ACL*.

Russo-Lassner, G., Lin, J., and Resnik, P. (2006). A paraphrase-based approach to machine translation evaluation. Technical report, University of Maryland, College Park.

Shinyama, Y., Sekine, S., Sudo, K., and Grishman, R. (2002). Automatic paraphrase acquisition from news articles. In *Proceedings of Human Language Technology Conference (HLT 2002)*, pages 313–318, San Diego, USA.

Siddharthan, A. (2002). An architecture for a text simplification system. In *Language Engineering Conference*, page 64. IEEE Computer Society.

Siddharthan, A. (2006). Syntactic simplification and text cohesion. *Research on Language and Computation*, 4(1):77–109.

Siddharthan, A. (2011). Text simplification using typed dependencies: a comparison of the robustness of different generation strategies. In *Proceedings of the 13th European Workshop on Natural Language Generation*, ENLG '11, pages 2–11, Stroudsburg, PA, USA. Association for Computational Linguistics.

Smith, D. A. and Eisner, J. (2006). Quasi-synchronous grammars: Alignment by soft projection of syntactic dependencies. In *Proceedings of the HLT-NAACL Workshop on Statistical Machine Translation*, pages 23–30, New York.

Snover, M., Madnani, N., Dorr, B., and Schwartz, R. (2010). Ter-plus: paraphrase, semantic, and alignment enhancements to translation edit rate. *Machine Translation*, 23(2-3):117–127.

Snover, M., Madnani, N., Dorr, B. J., and Schwartz, R. (2009). Fluency, adequacy, or HTER?: exploring different human judgments with a tunable MT metric. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, StatMT '09, pages 259–268, Stroudsburg, PA, USA. Association for Computational Linguistics.

Stolcke, A. (2002). SRILM - an extensible language modeling toolkit. In *In Proc. Int. Conf. on Spoken Language Processing*, pages 901–904, Denver, Colorado.

Theune, M., Koolen, R., Krahmer, E., and Wubben, S. (2011). Does size matter - how much data is required to train a reg algorithm? In *ACL (Short Papers)*, pages 660–664. The Association for Computer Linguistics.

Tiedemann, J. (2009). Character-based PSMT for closely related languages. In Marqués, L. and Somers, H., editors, *Proceedings of 13th Annual Conference of the European Association for Machine Translation (EAMT'09)*, pages 12 – 19, Barcelona, Spain.

Turner, J. and Charniak, E. (2005). Supervised and unsupervised learning for sentence compression. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 290–297, Ann Arbor, Michigan.

Van Erp, M., Van Den Bosch, A., Wubben, S., and Hunt, S. (2009). Instance-driven discovery of ontological relation labels. In *Proceedings of the EACL 2009 Workshop on Language Technology and Resources for Cultural Heritage, Social Sciences, Humanities, and Education*, pages 60–68. Association for Computational Linguistics.

Van Huyssteen, G. B. and Pilon, S. (2009). Rule-based conversion of closely-related languages: a dutch-to-afrikaans convertor. *20th Annual Symposium of the Pattern Recognition Association of South Africa*.

Vandeghinste, V. and Pan, Y. (2004). Sentence compression for automated subtitling: A hybrid approach. In *Proceedings of the ACL Workshop on Text Summarization*, pages 89–95.

Vandeghinste, V. and Tjong Kim Sang, E. (2004). Using a Parallel Transcript/Subtitle Corpus for Sentence Compression. In *Proceedings of LREC 2004*.

Vickrey, D. and Koller, D. (2008). Sentence simplification for semantic role labeling. In *Proceedings of the 46th Meeting of the Association for Computational Linguistics: Human Language Technologies*.

Vilar, D., Peter, J.-T., and Ney, H. (2007). Can we translate letters? In *Second Workshop on Statistical Machine Translation*, pages 33–39, Prague, Czech Republic. Association for Computational Linguistics.

Voorhees, E. (2001). Question answering in TREC. In *Proceedings of the tenth international conference on Information and knowledge management*, pages 535–537. ACM.

Vossen, P., Maks, I., Segers, R., and VanderVliet, H. (2008). Integrating lexical units, synsets and ontology in the cornetto database. In Nicoletta Calzolari (Conference Chair), K. C., editor, *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2008/.

Watanabe, W. M., Junior, A. C., de Uzêda, V. R., de Mattos Fortes, R. P., Pardo, T. A. S., and Aluísio, S. M. (2009). Facilita: reading assistance for low-literacy readers. In Mehlenbacher, B., Protopsaltis, A., Williams, A., and Slattery, S., editors, *SIGDOC*, pages 29–36. ACM.

Woodsend, K. and Lapata, M. (2011). Learning to simplify sentences with quasi-synchronous grammar and integer programming. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 409–420, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Wubben, S. (2010). UvT: Memory-based pairwise ranking of paraphrasing verbs. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 260–263. Association for Computational Linguistics.

Wubben, S., Marsi, E., van den Bosch, A., and Krahmer, E. (2011a). Comparing phrase-based and syntax-based paraphrase generation. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, pages 27–33. Association for Computational Linguistics.

Wubben, S. and van den Bosch, A. (2009). A semantic relatedness metric based on free link structure. In *Proceedings of the Eighth International Conference on Computational Semantics*, IWCS-8 '09, pages 355–358, Stroudsburg, PA, USA. Association for Computational Linguistics.

Wubben, S., van den Bosch, A., and Krahmer, E. (2010). Paraphrase generation as monolingual translation: Data and evaluation. In J. Kelleher, B. M. N. and van der Sluis, I., editors, *Proceedings of the 10th International Workshop on Natural Language Generation (INLG 2010)*, pages 203–207, Dublin.

Wubben, S., van den Bosch, A., and Krahmer, E. (2011b). Paraphrasing headlines by machine translation: Sentential paraphrase acquisition and generation using Google News. *Computational Linguistics in the Netherlands 2010: Selected Papers from the Twentieth CLIN Meeting*, pages 169–183.

Wubben, S., van den Bosch, A., and Krahmer, E. (2012). Sentence simplification by monolingual machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1015–1024, Jeju Island, Korea. Association for Computational Linguistics.

Wubben, S., van den Bosch, A., Krahmer, E., and Marsi, E. (2009). Clustering and matching headlines for automatic paraphrase acquisition. In Krahmer, E. and Theune, M., editors, *The 12th European Workshop on Natural Language Generation*, pages 122–125, Athens. Association for Computational Linguistics.

Xu, W., Ritter, A., Dolan, B., Grishman, R., and Cherry, C. (2012). Paraphrasing for style. In *COLING*, pages 2899–2914.

Yamada, K. and Knight, K. (2001). A syntax-based statistical translation model. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, ACL '01, pages 523–530, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yatskar, M., Pang, B., Danescu-Niculescu-Mizil, C., and Lee, L. (2010). For the sake of simplicity: Unsupervised extraction of lexical simplifications from Wikipedia. In *Proceedings of the NAACL*, pages 365–368.

Zajic, D., Dorr, B. J., Lin, J., and Schwartz, R. (2007). Multi-candidate reduction: Sentence compression as a tool for document summarization tasks. *Information Processing Management*, 43(6):1549–1570.

Zeldes, A. (2007). Machine translation between language stages: Extracting historical grammar from a parallel diachronic corpus of Polish. In Davies, M., Rayson, P., Hunston, S., and Danielsson, P., editors, *Proceedings of the Corpus Linguistics Conference CL2007*. University of Birmingham.

Zens, R., Och, F. J., and Ney, H. (2002). Phrase-based statistical machine translation. In *Proceedings of the 25th Annual German Conference on AI: Advances in Artificial Intelligence*, KI '02, pages 18–32, London, UK. Springer-Verlag.

Zhao, S., Lan, X., Liu, T., and Li, S. (2009). Application-driven statistical paraphrase generation. 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: Volume 2 - Volume 2*, ACL '09, pages 834–842, Stroudsburg, PA, USA. Association for Computational Linguistics.

Zhou, L., Lin, C.-Y., and Hovy, E. (2006). Re-evaluating machine translation results with paraphrase support. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 77–84, Sydney, Australia. Association for Computational Linguistics.

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 (Coling 2010)*, pages 1353–1361, Beijing, China. Coling 2010 Organizing Committee.

LIST OF FIGURES

LIST OF TABLES

SUMMARY

Text-to-text generation can be defined as the rewriting of a text according to certain requirements. Performing automatic monolingual text-to-text generation can be instrumental in solving many natural language processing problems. Paraphrase generation can for instance help increase the range of applications such as question answering.

Since the monolingual text-to-text generation field is relatively new, no established methods exist yet. The challenges faced in the research area of machine translation are however similar to the challenges in monolingual text-to-text generation. Both disciplines are involved in generating grammatically correct output and in generating output that is meaningfully related to the input of the system. An interesting venue for research is then the application of established methods in machine translation to diverse text-to-text generation tasks. The first research question we answer in this thesis is:

1. How can a statistical machine translation model be applied to a collection of monolingual text-to-text generation tasks?

Although generally corpora for machine translation are abundant, the corpora that are available for monolingual text-to-text generation are often relatively small and most of the data is English. The second research question we pose is then:

2. How can good parallel monolingual corpora be created?

In addition to the challenge of the acquisition of data, there is the issue of evaluation. One factor in making successful text-to-text applications is a proper evaluation methodology. This leads to the following research question:

3. To what extent can text-to-text generation be evaluated automatically?

In each chapter we discuss a different text-to-text generation task. Our research starts In Chapter 2, in which we investigate the automatic generation of paraphrases by using machine translation techniques. Three contributions we make are the construction of a sufficiently large paraphrase corpus, a re-ranking heuristic to use machine translation for paraphrase generation and a proper evaluation methodology. A large parallel corpus is constructed by aligning clustered headlines that are crawled from a news aggregator site. To generate sentential paraphrases we use a standard phrase-based machine translation (PBMT) framework modified with a re-ranking component (PBMT-R). We demonstrate this approach for Dutch and English and evaluate by using human judgements collected from 76 participants. The judgments are compared to two automatic machine translation evaluation metrics. We observe that as the paraphrases deviate more from the source sentence, the performance of the PBMT-R system degrades less than that of the word substitution baseline system.

In Chapter 3 we describe a method for simplifying sentences using the PBMT-R model, trained on a parallel simplification corpus. We compare our system to a word-substitution baseline and two state-of-the-art systems, all trained and tested on paired sentences from the English part of Wikipedia and Simple Wikipedia. Human test subjects judge the output of the different systems. Analysis of the judgements shows that by relatively careful phrase-based paraphrasing our model achieves similar simplification results to state-of-the-art systems, while generating better formed output. We also argue that text readability metrics such as the Flesch-Kincaid grade level should be used with caution when evaluating the output of simplification systems.In this chapter we will discuss the task of sentence compression.

In Chapter 4 we present a memory-based approach that can perform sentence compression by deletion (extractive compression) and a hybrid model that makes use of phrase-based machine translation that in addition to deletions can also perform compressions by paraphrasing longer source phrases into shorter target phrases (abstractive compression). Because no sufficiently large abstractive compression corpora exist, we train the phrase-based machine translation component of the hybrid model on simplification data. We will describe the extractive and abstractive systems and let human judges evaluate the output of these systems. Although in general we expect humans to evaluate abstractive compression more positive than extractive compression, abstractive compression is a task that is considerably more difficult than extractive compression, and there is no abstractive data available. We therefore expect the extractive approach to be a strong approach.

In this Chapter 5 we investigate language transformation. Language transformation can be defined as translating between diachronically distinct lan-

guage variants. We investigate the transformation of Middle Dutch into Modern Dutch by means of machine translation. For diachronic language transformation we have to rely on parallel data that has been made available consisting of paired sentences from the two diachronic variants of the language. This means generally not much data is available. We tackle this problem by making use of the characteristics of the data. We demonstrate that by using character overlap the performance of the machine translation process can be improved for this task.

We end the thesis with Chapter 6, in which we discuss the results and answer the research questions.

PUBLICATIONS

- Wubben, S., van den Bosch, A., Krahmer, E., and Marsi, E. (2009). Clustering and matching headlines for automatic paraphrase acquisition. In Krahmer, E. and Theune, M., editors, *The 12th European Workshop on Natural Language Generation*, pages 122–125, Athens. Association for Computational Linguistics

- Wubben, S. and van den Bosch, A. (2009). A semantic relatedness metric based on free link structure. In *Proceedings of the Eighth International Conference on Computational Semantics*, IWCS-8 '09, pages 355–358, Stroudsburg, PA, USA. Association for Computational Linguistics

- Van Erp, M., Van Den Bosch, A., Wubben, S., and Hunt, S. (2009). Instance-driven discovery of ontological relation labels. In *Proceedings of the EACL 2009 Workshop on Language Technology and Resources for Cultural Heritage, Social Sciences, Humanities, and Education*, pages 60–68. Association for Computational Linguistics

- Wubben, S., van den Bosch, A., and Krahmer, E. (2010). Paraphrase generation as monolingual translation: Data and evaluation. In J. Kelleher, B. M. N. and van der Sluis, I., editors, *Proceedings of the 10th International Workshop on Natural Language Generation (INLG 2010)*, pages 203–207, Dublin

- Wubben, S. (2010). UvT: Memory-based pairwise ranking of paraphrasing verbs. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 260–263. Association for Computational Linguistics

- Wubben, S., van den Bosch, A., and Krahmer, E. (2011b). Paraphrasing headlines by machine translation: Sentential paraphrase acquisition and generation using Google News. *Computational Linguistics in*

the Netherlands 2010: Selected Papers from the Twentieth CLIN Meeting, pages 169–183

- Theune, M., Koolen, R., Krahmer, E., and Wubben, S. (2011). Does size matter - how much data is required to train a reg algorithm? In *ACL (Short Papers)*, pages 660–664. The Association for Computer Linguistics

- Wubben, S., Marsi, E., van den Bosch, A., and Krahmer, E. (2011a). Comparing phrase-based and syntax-based paraphrase generation. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, pages 27–33. Association for Computational Linguistics

- Wubben, S., van den Bosch, A., and Krahmer, E. (2012). Sentence simplification by monolingual machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1015–1024, Jeju Island, Korea. Association for Computational Linguistics

TICC PHD SERIES

1. Pashiera Barkhuysen. *Audiovisual Prosody in Interaction.* Promotores: M.G.J. Swerts, E.J. Krahmer. Tilburg, 3 October 2008.
2. Ben Torben-Nielsen. *Dendritic Morphology: Function Shapes Structure.* Promotores: H.J. van den Herik, E.O. Postma. Co-promotor: K.P. Tuyls. Tilburg, 3 December 2008.
3. Hans Stol. *A Framework for Evidence-based Policy Making Using IT.* Promotor: H.J. van den Herik. Tilburg, 21 January 2009.
4. Jeroen Geertzen. *Dialogue Act Recognition and Prediction.* Promotor: H. Bunt. Co- promotor: J.M.B. Terken. Tilburg, 11 February 2009.
5. Sander Canisius. *Structured Prediction for Natural Language Processing.* Promotores: A.P.J. van den Bosch, W. Daelemans. Tilburg, 13 February 2009.
6. Fritz Reul. *New Architectures in Computer Chess.* Promotor: H.J. van den Herik. Co- promotor: J.W.H.M. Uiterwijk. Tilburg, 17 June 2009.
7. Laurens van der Maaten. *Feature Extraction from Visual Data.* Promotores: E.O. Postma, H.J. van den Herik. Co-promotor: A.G. Lange. Tilburg, 23 June 2009 (cum laude).
8. Stephan Raaijmakers. *Multinomial Language Learning.* Promotores: W. Daelemans, A.P.J. van den Bosch. Tilburg, 1 December 2009.
9. Igor Berezhnoy. *Digital Analysis of Paintings.* Promotores: E.O. Postma, H.J. van den Herik. Tilburg, 7 December 2009.
10. Toine Bogers. *Recommender Systems for Social Bookmarking.* Promotor: A.P.J. van den Bosch. Tilburg, 8 December 2009.

11. Sander Bakkes. *Rapid Adaptation of Video Game AI.* Promotor: H.J. van den Herik. Co-promotor: P. Spronck. Tilburg, 3 March 2010.

12. Maria Mos. *Complex Lexical Items. Promotor: A.P.J. van den Bosch.* Co-promotores: A. Vermeer, A. Backus. Tilburg, 12 May 2010 (in collaboration with the Department of Language and Culture Studies).

13. Marieke van Erp. *Accessing Natural History. Discoveries in data cleaning, structuring, and retrieval.* Promotor: A.P.J. van den Bosch. Co-promotor: P.K. Lendvai. Tilburg, 30 June 2010.

14. Edwin Commandeur. *Implicit Causality and Implicit Consequentiality in Language Comprehension.* Promotores: L.G.M. Noordman, W. Vonk. Co-promotor: R. Cozijn. Tilburg, 30 June 2010.

15. Bart Bogaert. *Cloud Content Contention.* Promotores: H.J. van den Herik, E.O. Postma. Tilburg, 30 March 2011.

16. Xiaoyu Mao. *Airport under Control.* Promotores: H.J. van den Herik, E.O. Postma. Co-promotores: N. Roos, A. Salden. Tilburg, 25 May 2011.

17. Olga Petukhova. *Multidimensional Dialogue Modelling.* Promotor: H. Bunt. Tilburg, 1 September 2011.

18. Lisette Mol. *Language in the Hands.* Promotores: E.J. Krahmer, F. Maes, M.G.J. Swerts. Tilburg, 7 November 2011 (cum laude).

19. Herman Stehouwer. *Statistical Language Models for Alternative Sequence Selection.* Promotores: A.P.J. van den Bosch, H.J. van den Herik. Co-promotor: M.M. van Zaanen. Tilburg, 7 December 2011.

20. Terry Kakeeto-Aelen. *Relationship Marketing for SMEs in Uganda.* Promotores: J. Chr. van Dalen, H.J. van den Herik. Co-promotor: B.A. Van de Walle. Tilburg, 1 February 2012.

21. Suleman Shahid. *Fun & Face: Exploring non-verbal expressions of emotion during playful interactions.* Promotores: E.J. Krahmer, M.G.J. Swerts. Tilburg, 25 May 2012.

22. Thijs Vis. *Intelligence, Politie en Veiligheidsdienst: Verenigbare Grootheden?* Promotores: T.A. de Roos, H.J. van den Herik, A.C.M. Spapens. Tilburg, 6 June 2012 (in collaboration with the Tilburg School of Law).

23. Nancy Pascall. *Engendering Technology Empowering Women.* Promotores: H.J. van den Herik, M. Diocaretz. Tilburg, 19 November, 2012.

24. Agus Gunawan. *Information Access for SMEs in Indonesia.* Promotor: H.J. van den Herik. Co-promotores: M. Wahdan, B.A. Van de Walle. Tilburg, 19 December 2012.

25. Giel van Lankveld. *Quantifying Individual Player Differences.* Promotores: H.J. van den Herik, A.R. Arntz. Co-promotor: P. Spronck. Tilburg, 27 February 2013.

26. Sander Wubben. *Text-to-Text Generation by Monolingual Machine Translation.* Promotores: E.J. Krahmer, A.P.J. van den Bosch, H. Bunt. Tilburg, 5 June 2013.

SIKS PHD 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. E.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 Coupâ (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*
4. Evgueni Smirnov (UM) *Conjunctive and Disjunctive Version Spaces with Instance-Based Boundary Sets*
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*

8. Pascal van Eck (VU) *A Compositional Semantic Structure for Multi-Agent Systems Dynamics.*

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 forwork practice analysis and design*

11. Tom M. van Engers (VUA) *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*

3. Henk Ernst Blok (UT) *Database Optimization Aspects for Information Retrieval*

4. Juan Roberto Castelo Valdueza (UU) *The Discrete Acyclic Digraph Markov Model in Data Mining*

5. Radu Serban (VU) *The Private Cyberspace Modeling Electronic Environments inhabited by Privacy-concerned Agents*

6. Laurens Mommers (UL) *Applied legal epistemology; Building a knowledge-based ontology of the legal domain*

7. Peter Boncz (CWI) *Monet: A Next-Generation DBMS Kernel For Query-Intensive Applications*

8. Jaap Gordijn (VU) *Value Based Requirements Engineering: Exploring Innovative E-Commerce Ideas*

9. Willem-Jan van den Heuvel(KUB) *Integrating Modern Business Applications with Objectified Legacy Systems*

10. Brian Sheppard (UM) *Towards Perfect Play of Scrabble*

11. Wouter C.A. Wijngaards (VU) *Agent Based Modelling of Dynamics: Biological and Organisational Applications*

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*

2. Jan Broersen (VU) *Modal Action Logics for Reasoning About Reactive Systems*

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*

10. Andreas Lincke (UvT) *Electronic Business Negotiation: Some experimental studies on the interaction between medium, innovation context and culture*

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 politiâ°le gegevensuitwisseling en digitale expertise*

9. Martin Caminada (VU) *For the Sake of the Argument; explorations into argument-based reasoning*

10. Suzanne Kabel (UVA) *Knowledge-rich indexing of learning-objects*
11. Michel Klein (VU) *Change Management for Distributed Ontologies*
12. The Duy Bui (UT) *Creating emotions and facial expressions for embodied agents*
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*
18. Bart Orriens (UvT) *On the development an management of adaptive business collaborations*
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 Ramirez Camps (CWI) *Structural Features in XML Retrieval*
25. Joost Schalken (VU) *Empirical Investigations in Software Process Improvement*

2008

1. Katalin Boer-Sorban (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 learning algorithms*
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*

5. Marijn Plomp (UU) *Maturing Interorganisational Information Systems*

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*

31. Emily Bagarukayo (RUN) *A Learning by Construction Approach for Higher Order Cognitive Skills Improvement, Building Capacity and Infrastructure*

32. Wietske Visser (TUD) *Qualitative multi-criteria preference representation and reasoning*

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*

39. Hassan Fatemi (UT) *Risk-aware design of value and coordination networks*

40. Agus Gunawan (UvT) *Information Access for SMEs in Indonesia*

41. Sebastian Kelle (OU) *Game Design Patterns for Learning*

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*

46. Simon Carter (UVA) *Exploration and Exploitation of Multilingual Data for Statistical Machine Translation*

47. Manos Tsagkias (UVA) *Mining Social Media: Tracking Content and Predicting Behavior*

48. Jorn Bakker (TUE) *Handling Abrupt Changes in Evolving Time-series Data*

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*

3. Szymon Klarman (VU) *Reasoning with Contexts in Description Logics*

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 (UT) *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. Maryam Razavian (VU) *Knowledge-driven Migration to Services*

13. Mohammad Zafiri (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 Steenhuisen (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*