Paraphrasing In The Context Of Computer Aided Translation

Turan Rustamli

Master of Science
Artificial Intelligence
School of Informatics
University of Edinburgh
2013

Abstract

...

i

Acknowledgements

Many thanks.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Turan Rustamli*)

Table of Contents

Chapter 1

Introduction

1.1 Motivation

Today, taking advantage of advancements in machine translation, we can read and understand text in languages unknown to us. However, producing a high quality translation is a task that is still solved by humans. Computer Aided Translation (CAT) systems aim to support and assist translators, providing them with tools that partially automate and simplify their workflow. Demand for these systems is huge, each year billions are spent on translation by international organisations such as European Union.

Several assistance ways, that could be offered by a CAT system, were introduced since investigation of the question began. The most popular ones are sentence-completion and post-editing. While the first approach is providing suggestions while user is typing translation, the second approach prefills text input area with a machine translation.

In this report we present a novel assistance way that aims to help translator to fix an initial machine generated translation, by providing paraphrases for any user-specified part of the sentence.

Paraphrasing is the process of representing the same idea with different words. Some recent publications suggest that paraphrasing and translation tasks are closely related and information used to perform one task, can be useful for another (ccb). Considering this we are introducing paraphrasing approach that efficiently reuses the information available after machine translation process which output is used to provide post-editing assistance within a CAT system.

Moreover, taking advantage of interactivity of a CAT process, we investigated paraphrasing as an interactive task, implementing a system that accepts and responds to user's feedback on the paraphrases that were initially produced.

We will also describe a novel automatic evaluation approach that we used to test feasibility and usefulness of paraphrasing results. Outcomes of a set of user studies will be provided to back up the results of automatic evaluation.

1.2 Overview

In this section we would like to provide an outline of the report.

We will start the following chapter by introducing background material in the next chapter. Recent advancements in CAT will be presented and references to previous publications about paraphrasing will be provided. We will also discuss some basis concepts of machine translation, particularly Search Graph representation, which will be used by us to solve paraphrasing task.

In the next chapter we will focus on providing description of work that was done in scope of our project. First, we will start by discussing design of our approach and explaining it's possible ways of integration with existing CAT systems. We will continue by providing details of implementation of our design. Description of challenging problems, that were solved during design and implementation stages, and their solutions will be given as well.

We will present our evaluation methodology, stating results of testing various versions of our paraphrasing algorithm in the Chapter 4. This includes description of an automatic evaluation framework which was implemented by us during the project, it's advantages and disadvantages. Next, we will provide results of a set of user studies carried out by us in order to support results of automatic evaluation.

Following chapter will introduce novel experiments by making paraphrasing within a CAT system more interactive. Specifically, we will present a design and implementation of a feature that lets a user to instantly interact with the system and provide improved results. In this chapter we will discuss "Show More Like This" button that was added by us to the paraphrases list displayed to a user, which is similar to ideas of relevance feedback () in Information Retrieval in order to provide user with more relevant paraphrases. A modification that was made to our automatic evaluation framework in order to support this new feature will be also introduced, as well as results of both automatic and manual evaluation.

Next, we will look into another way of adding more interactivity to the user interface of the paraphrasing functionality, by implementing and adaptive paraphrase button that will let user to initially suggest us the reason of the paraphrasing request. We will

discuss situations when users might find paraphrasing functionality useful interacting with a CAT system, providing results of user studies.

In our final chapter we will conclude our report and suggest interesting directions for the future work.

Chapter 2

Background

The main idea of our project is providing a paraphrasing feature within a Computer Aided Translation system by reusing the information that is available as a result of machine translation process which output is used to provide user with an initial translation. Before introducing our approach, we would like to provide some background information about CAT systems, relationship between paraphrasing and translation tasks as well as some concepts of machine translation that were are used by our approach to implement paraphrasing.

2.1 Computer Aided Translation

Recently, more and more papers discussing various aspects of Computer Aided Translation are published. This increase could be explained by the fact that the need of high quality and publishable translation is still satisfied by human translators and how advancements in machine translation are efficiently used by them in order to increase productivity of translators is still an open question. Various ways of assistance were suggested and implemented. Most of these tools are reportedly increasing translation speed, others help to achieve a better quality. In this work we present a novel assistance type, which aims to help translators by providing with paraphrases - alternative ways of expressing the same ideas. Before discussing paraphrasing, we want to highlight the ways our approach could be used in the context of CAT systems that provide different types of assistance.

2.1.1   Interactive Machine Translation

One way to assist translators is providing auto-completion suggestions while he composes the translation in target language. As user's input in text editor component changes, system suggests possible translation of the following word or phrase. The first system implementing this approach was TransType (Langlais et al., 2000; Foster et al., 2002; Bender et al., 2005), providing optional sentence-completion suggestions based on machine translation. Based on user's acceptance of the suggestions, system can be improved to provide better results (Barrachina). Caitra is another CAT system described by Koehn and Haddow (2009) which also provides an interactive mode, where system uses search graph of machine translation decoder to suggest multiple options for upcoming words and phrases translation.

Our paraphrasing approach similarly reuses search graph for generation of alternative translation options, however in contrast to Interactive Machine Translation systems we assume that the user already has an initial translation, that she wants to improve using better wording. This initial translation could be originally produced using an interactive assistance with auto-completion or alternatively it could be a result provided for post-editing.

2.1.2   Post-Editing

Another popular assistance way is providing machine translation for post-editing by translator. In this case for each sentence in the document that is being translated, multiple translation options are provided. User can select one by clicking on it, what will update value of text editor component with the chosen translation. Alternatively, user can ignore suggested options and start typing translation manually and possibly she can take advantage of an aforementioned interactive suggestions.

Effect of post-editing on productivity of the translation process was analysed by many researchers, including (...). It was previously demonstrated that post-editing can significantly improve performance of translator in some contexts like legal and information technology documents (...). However, in other domains like news and media, initial translation produced for post-editing is far from publishable quality. This output is being manually fixed and improved, by replacing erroneous parts with correct words preserving the meaning. This is one of the situations when our assistance paraphrasing service aims to help user to find multiple possible alternative translations of the corrupted part.

During this project we were investigating feasibility of our tool, in scope of a CAT system that provides post-editing options. However, our approach requires only information generated during machine translation decoding, the search graph, which is also being build during interactive translation. That means our technique could be easily extended to be used with the CAT systems without post-editing, that provide machine translation driven assistance by other means.

2.1.3   Bilingual concordancer

Bilingual concordancer is a feature used by professional users to explore alternative translations of particular phrases. This is achieved by finding occurrences of the phrase in bilingual corpus and retrieving their corresponding translations. Analyses of usage logs of TransSearch, one of the most popular bilingual concordancers, for 6 years demonstrate that tranlators find this feature useful especially for finding the correct sense in case of highly polysemous adverbials and prepositional phrases Macklovitch et al. (2008). We believe that our paraphrasing tool that given one possible translation of a phrase explorers other alternative translations might be also useful in terms of finding a translation that preserves the meaning of the original phrase in a better way. We expect that starting the search within the same language, will simplify the this task for users with poor understanding of source language.

2.1.4   Other assistance ways

Providing various translation options for parts of sentence in source language, can also help in translation task. As it was shown by Koehn (2010), using this assistance it is possible to achieve translation with acceptable quality without understanding source language. The difference of our approach is that instead of providing different translation options for parts of original text, we produce paraphrasing options for any user specified parts of translation. This similarly to translation options, our tool can be used by someone, who doesn't know original language, to understand unclear parts of the text provided for post-editing, by paraphrasing them into representations that make more sense to him.

Finally, feasibility of our approach could be improved when used together with other CAT features like idioms and terminology detections. Indeed, having idioms or terminology marked in user's input for paraphrasing they could be treated in specific ways. In case of terminology paraphrasing lookup might take benefits of domain spe-

cific thesauruses. For idioms, paraphrasing algorithm will consider them as atomic units and will not try to paraphrase them partially.

2.1.5 Summary

In conclusion, we described how assisting translator by providing paraphrases of translation could be used together with other assistance means provided by a CAT system. While our implementation requires data generated by a translation decoder, this data is not supposed to be generated separately specifically for paraphrasing needs, but rather data available after running translation system for an interactive assistance or post-editing could be successfully reused for paraphrasing. We also compared our approach with tools like bilingual concordancer and translation options provider, highlighting similarities and differences. And finally, we mentioned how paraphrasing could take advantage of idiom and terminology detectors.

2.2 Paraphrasing Approaches

Here we will summarize outcomes of this and variety of other data-driven paraphrasing techniques previously available in the literature. Essential difference of our approach from the ones listed in this section is that while most of previous studies were focused on paraphrasing generation problem, trying to extract paraphrases from various sources and store them for future usage, our approach is considering paraphrasing as search problem and aims to find paraphrases for a phrase that is given in a context of a sentence that is being translated by a user. Despite this difference ideas from these approaches were used by us for achieving better results and for developing an automatic evaluation approach that was useful for assessing multiple versions of our paraphraser.

2.2.1 Data-driven paraphrasing

Originally paraphrasing was considered within multiple problems in Natural Language Processing, including question answering, summarization and generation. As a result historically there were multiple approaches for paraphrasing including techniques using formal semantic representation and methods that used grammars. More recent approaches are statistically motivated and make use of various data sources to generate paraphrases. Main data sources that were used in previous literature include multiple translations (for example multiple translations of classical literature), comparable

corpora (news articles about same events, different encyclopedia articles with same subject), and monolingual corpora (in this case distributional similarity is exploited to find paraphrases). A novel technique introduced by Callison-Burch (2007), describes how bilingual parallel corpora could be successfully used for paraphrase generation, this approach is related to the technique we use for paraphrasing. Similarly, the data we exploit is build during a statistical machine translation process and derives from a bilingual parallel corpora.

2.2.2 Multiple translations

The key idea of the techniques that exploit multiple translations as a data source for paraphrase generation is that translators composing different translation of the same text were preserving it's original meaning, what makes these translations natural sources for paraphrasing (). First experiments with multiple translation of French classical literature were focused on extracting paraphrases by detecting different phrases in similar contexts. So for example, following French sentence, might be translated into English in two different ways as follows:

Gènèralement, les gens qui savant peu parlent becoup, et les gens qui savant beaucoup parlent peu

(Jean-Jacques Rousseau)

People who know little speak a lot, and the people who know a lot speak little.

People who know little are usually great talkers, while men who know much say little.

Aligning similar parts of both sentences, we can see that "are usually great talkers" and "speak a lot" appear in same context and therefore they probably have same meaning.

Later researchers applied more complex ways to detect paraphrases in multiple translations, this includes using parse trees and considering parts of sentence with similar syntactic role to be paraphrases.

While our paraphrasing approach doesn't exploit this ideas. We used similar technique to develop an automatic paraphrasing evaluation framework. Instead of multiple

translations that are rarely available we used a machine translation to corrupt natural language sentences, and later used difference between original and corrupted sentences as paraphrasing target. This technique similarly decides whether parts represent same idea or not by considering the surrounding context. Our evaluation methodology will be discussed in detail in Chapter 5.

2.2.3 Bilingual parallel corpus

Close ties between paraphrasing and translation were studied by Callison-Burch (2007). The results of this work were important for designing our paraphrasing approach.

In contrast to previous studies that use monolingual data as main source for paraphrase detection, technique described by Callison-Burch (2007) exploits parallel corpus that contain text in two languages. Previously, this data has been considered mostly for solving statistical translation task. Authors demonstrated that similar statistical approach could be applied for solving paraphrasing problem as well.

In case of multiple translations and comparable corpora aligned equivalent sentences were detected and used as natural source for paraphrasing. However, this technique that deals with sentences paired with their translations, uses an alternative approach. It finds paraphrases using multiple occurrences of same foreign phrase that has different translations as pointers to possible paraphrases. Indeed, if two phrases have same translation in a given number of cases, they are probably encoding same meaning.

The main idea could be illustrated in terms of probabilities. Let's define paraphrase probability as $p(e_2|e_1)$, which designates a probability of that phrase e_2 is suitable paraphrase for given phrase e_1. This probability is defined similarly to the translation model probability $p(f|e_1)$, which expresses probability of that foreign phrase f has same meaning as given original phrase e_1. Another required expression is $p(e_2|f)$, the probability of that phrase e_2 is translation of given foreign phrase f. Considering that original phrase e_1 that we are trying to paraphrase may have multiple foreign translations, for the final expression paraphrase probability we will sum over f.

$$\hat{e}_2 = \arg \max_{e_2 \neq e_1} p(e_2|e_1) \tag{2.1}$$

$$\hat{e}_2 = \arg\max_{e_2 \neq e_1} \sum_f p(e_2|f)p(f|e_1) \qquad (2.2)$$

Callison-Burch (2007)

In equations *(2.1)* and *(2.2)* $\hat{e}_2$ represents the most probable paraphrase for origi-nally given phrase e_1. Multiple ways of computing probabilities $p(e_2|f)$ and $p(f|e_1)$ are studied in context of phrase based translation, for example maximum likelihood estimation could be used in the following way:

$$p(f|e_1) = \frac{count(e_1,f)}{count(e_1)} \qquad (2.3)$$

$$p(e_2|f) = \frac{count(e_2,f)}{count(f)} \qquad (2.4)$$

Callison-Burch (2007)

Here $count(\bar{e},\bar{f})$ stands for number of times phrase $\bar{e}$ is aligned with foreign phrase $\bar{f}$, and $count(\bar{u})$ is number of total occurrences of original or foreign phrase $\bar{u}$.

There are several problems highlighted by authors of the approach. Designing our paraphrasing approach we came across similar issues. This could be explained by the fact that data source we use originates from bilingual parallel corpus. We applied variations of the refinements suggested by Callison-Burch (2007) for achieving better paraphrasing results.

Firstly, they introduce issues with words and phrases that have multiple senses. While the main idea for detecting phrases that share same meaning is that they should have same translation in foreign language, it can be argued that foreign words with multiple senses will probably have different translations for each sense in source lan-guage. An example provided in Callison-Burch (2007), demonstrates how French words *banque* (bank in sense of financial institution) and *rive* (bank in sense of river-bank) both can appear in resulting English translation as word *bank*, and therefore could be mistakenly considered paraphrases. This problem was fixed by a refinement that constrains word sense during the paraphrasing.

Another problem occurs when translation in contrast to source uses a non-direct reference or hypernyms in some part of the sentence. This reference could be easily understood within the context that it was used. However as surrounding context is not considered during paraphrase generation it may result suggesting wrong paraphrases like "this organisation" and "European Union". Other example, could be inaccurate paraphrasing of "President Obama" into "President of United States". Suggested solution for this issue is adding specific constraints to paraphrases for example considering syntactic category, and agreement.

One more addition to the process of ranking paraphrases, that was suggested by authors was using language model probability considering context. Our approach also uses language model probability as one of the features contributing to the rank of the paraphrase, to do so we substitute phrases with their paraphrases in original sentences and evaluate them using language model.

Finally, the quality of paraphrasing is linked to the alignment quality, which is also crucial for translation quality. Authors suggest that using of multiple parallel corpora can reduce effect of systematic misalignment.

2.2.4 Other data-driven paraphrasing approaches

Two other techniques described in this section are not directly linked to our paraphrasing implementation. However ideas and heuristics from these methods could be used in order to collect an alternative training data for our paraphraser, which due to it's modular design can support additional auxiliary data sources.

Comparable corpus approach is using texts that discuss same subject as source for paraphrasing. For example, news articles that describe same event are mentioning same ideas and concepts using different wording. Different example of comparable corpus could be articles from different encyclopedias defining same concept. Finding paraphrases in this cases is more challenging task than in case of previous approach we discussed. But in contrast to multiple translation such data source is easier to find. String distance feature is used to detect similar parts in these texts. So if algorithm spots high concentration of similar wording in two sentences from different texts, it considers the sentences to be equivalent.

Researchers also noticed that in case of news articles, first sentences usually tend to summarize the main body of the text. Considering this, they treat the first sentences from all texts separately, considering that they probably carry same meaning. A par-

allel corpus is build from resulting sentence pairs and later the paraphrasing task is solved as monolingual translation problem. The problem with this approach is that after pairing sentences even from large sources, the resulting parallel corpus is significantly smaller, this is explained by the fact that detecting comparable sentence pairs is a complex problem, without an efficient solution.

A paraphrasing generation technique that uses a *monolingual corpus* as data source is primarily based on distributional similarities of phrases. Considering availability of this type of data, researchers proposing the approach suggest that given enough data it is possible to detect paraphrases by analysing patterns in frequencies of given groups of words. This approach is based on Distributional Hypothesis (Harris, 1954), which was originally proposed for detecting synonyms.

2.2.5 Summary

In this section we listed previous paraphrasing efforts highlighting their key ideas. These approaches mainly focus on generation repositories of paraphrases, in contrast to our approach which aims to use data not for extraction, but rather for a targeted paraphrase search. Ideas from multiple translations approach were used by us in automatic evaluation framework construction, some of the refinements applied to the bilingual parallel corpora paraphrasing technique were also useful in terms of improving output of our paraphraser.

Moreover, analysing paraphrasing as a interactive search problem, it's possible to reuse some ideas from Information Retrieval field. In Chapter 7 we will introduce a paraphrasing workflow that exploits feedback from users and instantly returns improved results. This feature was implemented using relevance feedback technique that is used by some web search engines to help users to redefine their original queries.

2.3 Decoding and search graph

As it was mentioned earlier, paraphrasing approach presented in this report reuses information generated during machine translation. Particularly, it utilises data structure known as *search graph*, that is being built during the decoding stage of translation. Here we will briefly review main concepts of Statistical Machine Translation, the process of decoding and how it could represented using a search graph. This section will also motivate usage of this data structure for paraphrases search.

2.3.1 A brief introduction to Statistical Machine Translation

Statistical Machine Translation (SMT) is one of the fundamental approaches of machine translation, which currently is used to achieve state of the art performance in the field. The main idea is using a parallel corpus which contains aligned phrases in two languages to statistically learn how words and phrases in these languages are related to each other and later employing this information to produce translation of any given input from one language to another. This could be expressed using the following Bayes' Rule equation, which expresses best translation in target language of an input in source language:

$$\arg\max_{target} p(target|source) = \arg\max_{target} p(source|target)p(target) \quad (2.5)$$

Two main components of this equation are $p(source|target)$, known as *translation model* and $p(target)$, known as *language model*. First expresses the likelihood of a *source* (input in original language) being translation of *target* (part of text in foreign language), and second expresses odds of using *target* in foreign language. While bilingual parallel corpus is required to build translation model, monolingual corpus could be used to produce language model.

The leading SMT approach is knows as phrase-based machine translation. In this approach group of words (*phrases*) are used together with individual words in order to achieve the most likely translation. This phrases are not linguistically driven, but rather statistically extracted from corpus. The process of associating words and phrases of source language with corresponding words and phrases of foreign language is knows as word alignment. Various techniques were introduced to solve this problem, one of the most successful approaches is GIZA++ that is available within Moses a translation toolkit that we used during this project.

Word alignments are used to build translation model, which could represented as *phrase table*. This data structure is used to find best translation of given source sentence. This process is known as decoding and information collected during it is used by us as source for paraphrasing.
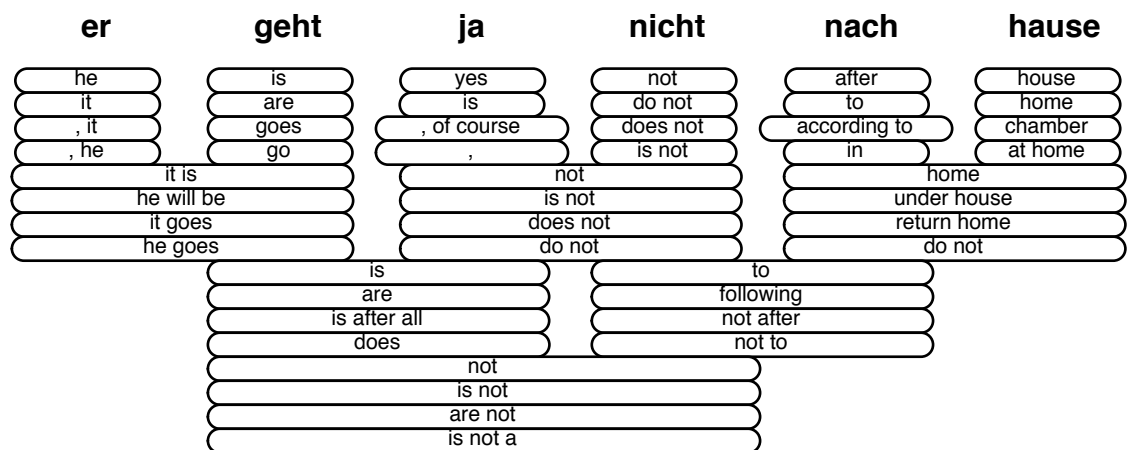
Figure 2.1: Translation Options

Koehn (2009)

2.3.2 Decoding

Given an input sentence, the number of possible translations grows exponentially [Koehn (2009)]. Even for short sentences it is not possible to score all possible ways of translation. Considering such a large search space it's unlikely to find the most probable translation. It was shown that decoding problem, given required machine translation models is an NP complete [[Knight (1999)]]. However, using various heuristics it is possible to find translation that is good enough for understanding the meaning of the sentence in source language. This is achieved by applying beam-search decoding algorithm that explorers a search space that is reduced by recombination and pruning techniques. While main goal of the process is producing one the most suitable output in target language, it is also possible to produce given number of candidate translations that are considered to be good enough by the search algorithm.

2.3.3 Search graph

By grouping words in the input sentence in different ways to form phrases, it is possible to achieve many different translations. Each phrase in source sentence has multiple corresponding translations, which are called *translation options*. Figure 2.1 demonstrates how sample German sentence can be separated into phrase in various ways, and depending on this separation phrases will have different English translation options.

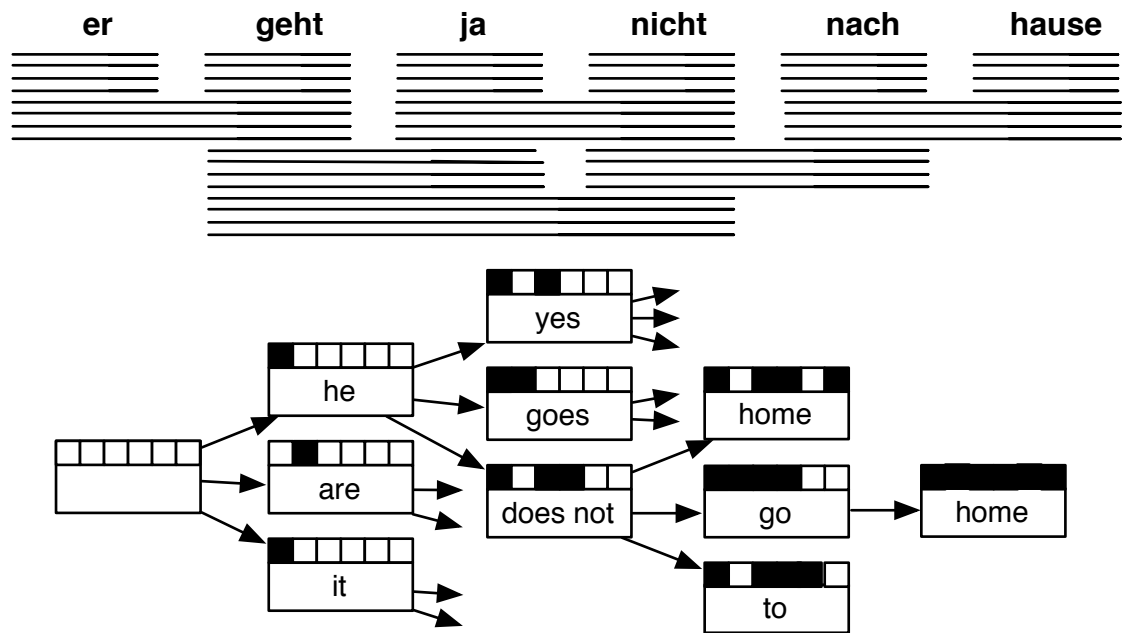   This translation options are used as building blocks in process known as *hypoth-*

Figure 2.2: Hypothesis expansion

Koehn (2009)

esis expansion. During this process, a data structure known as *search graph* is being built by sequentially joining translation options that cover different parts of sentence in source language (Figure 2.2). Nodes of this graph are called *hypotheses*. As new translation options are being added to existing hypotheses, new partial translation is formed Koehn (2009). This process continues until sequence of hypotheses cover all parts of input sentence. Each hypothesis contains information about the translation being attached in current step, total score (based on log-probability) of current partial translation, link to previous hypothesis that is extended by current one and change in coverage vector of the foreign language sentence.

2.3.4 Hypothesis recombination

The resulting data structure is still to large to be directly used for translation search. Several techniques are applied in order to reduce it's size. One way is using hypothesis recombination, which merges paths with same output and foreign language coverage. This reduction is risk-free and may not cause losing a good translation option. Sample of recombination is illustrated in Figures 2.3 and 2.4. Both hypotheses before recombination produce same output and cover same parts of source sentence, therefore the one
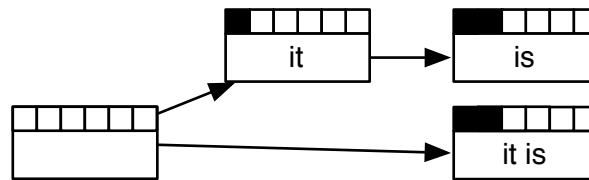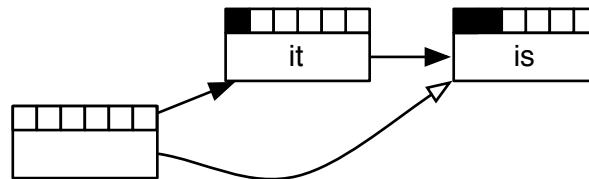
Figure 2.3: Before recombination

Koehn (2009)

Figure 2.4: After recombination

Koehn (2009)

with lower score could be recombined to reduce search space. Although this technique removes many unnecessary paths from graph, applying recombination doesn't affect overall exponential growth in number of hypothesis.

2.3.5   Pruning

Further reduction of search graph is required for longer sentences. *Stack Decoding* is applied in order to filter hypothesis that have low probability of leading to good translations. Essential difference from recombination is that, this reduction technique is not risk-free and potentially may eliminate some useful translations. During this process stacks containing hypotheses with same coverage are constructed and pruning methods like histogram and threshold pruning are applied. As result number of items in stacks are reduced.

2.3.6   Summary

In this section we provided a brief introduction into phrase-based Statistical Machine Translation, focusing on decoding stage during which search graph is being produced. Two main reasons of erroneous translation produced by the process described are

search error and *model error*. While the first one corresponds to failure of finding correct translation during search, second one refers to initial problems in machine translation models. Assistance tool developed and investigated by us aims to provide users with an option to fix these errors within a CAT system by pointing to erroneous parts and selecting a better paraphrasing option from the list retrieved by the system. In some sense our approach lets user to reuse information collected during decoding. This information is available as search graph output, which optionally could be returned by Moses, together with translation results.

Chapter 3

Design and implementation

In previous chapter, we introduced Computer Aided Translation, discussing how multiple assistance ways provided within systems implementing this concept relate to our novel tool that aims to support translators by providing an ability to fix specified parts of the translation using paraphrasing options. Furthermore, we discussed previous paraphrasing approaches highlighting similarities and difference with technique we use to search paraphrases. And finally a brief overview of phrase-based Statistical Machine Translation was provided to introduce origins of the search graph, data structure that we use as main source for paraphrasing.

Current chapter will provide detailed description of work that was carried out during this project. We start by analysing requirements for paraphrasing service within a CAT system. After that an iterational design and implementation process we applied to develop working version of such service. Major design decision that were made during the work on project will be introduced and justified.

3.1   Analysing requirements

As mentioned earlier, an essential aspect of our paraphrasing approach is that it aims to provide a real-time targeted paraphrase search. Unlike previous efforts discussed by us before, we consider an interactive environment were queries sent by users are executed against our data model and results after being dynamically filtered and ranked using various heuristics are returned to user interface. In this section we list main requirements for our approach that were considered during design stage.

Integration with existing CAT software. Taking into account that most of modern CAT implementations are providing web based service (Caitra, Casmacat, MateCat),

we also employed the client-server architecture to design our paraphrasing service. It might be easily attached as a module to the services provided online. Moreover, our service could also be hosted as a standalone service and integrated with desktop translation software by providing an API. This way our paraphrasing assistance might be accessible from all major types of CAT applications.

Reusing results of machine translation. Another important feature of our paraphrasing system is being able to produce paraphrases using only data available as a result of machine translation that provides post-editing assistance, rather requiring any additional paraphrase corpora. We designed our implementation to support translation output generated by Moses []. The process of retrieving and processing search graph will be reviewed by us in [section 3.2].

Importance of ranking. Quality of final ranking of the paraphrasing options is crucial requirement for RePhraser that aims to provide an interactive service. Indeed, basic concepts of Human-Computer Interaction suggest that paraphrasing options displayed to users will be efficient only in case if there count doesn't exceed 7. This means that it's important to rank top results in a way that they will contain at least one good paraphrase. Features that we exploited for ranking as well as a dynamic system of filters and sorters that we applied to improve top results will be described in [section 3.5].

Paraphrasing granularity. Various levels of paraphrase granularity are studied. In case of entire sentences the task is known *sentential* (or *clausal*) paraphrasing, for shorter items it is called *lexical* (or *phrasal*) paraphrasing. For our project we investigated only lexical paraphrases, motivating this decision by the fact that other assistance tools within a CAT system may already provide multiple translation options for entire sentence, for example results of translation by using different bilingual parallel corpora. We also acknowledge that within scope of lexical paraphrases there is a further separation into two classes that correspond to shorter and longer phrases. Both cases were considered by us during evaluation stage.

We also took into account following study [], which analyses usage patterns of another assistance tool, bilingual concordancer where authors suggest that average length of the input query is 2-3 words. Parallels between this assistance way and ours were discussed in previous chapter and in practice they motivate expectation of average paraphrasing query length to be similar.

Realtime experience. Another requirement for our interactive tool is providing paraphrasing options as fast as possible. Our tool will not be useful at all if it takes

it to return results longer that time user spends on manual paraphrasing. Thus we considered time required to generate paraphrases in our evaluation.

Coexistence with manual editing. While RePhraser aims to be used in order to fix erroneous part in the translation by getting the corrected paraphrase automatically, we acknowledge that final high quality translation cannot be achieved only by using paraphrasing. An ability to manually post-edit translation is crucial and designing our approach we considered that it will be used in an environment where translation could be altered by user at any time.

3.2 Preparing data

3.2.1 Retrieving search graph dump file

A representation of the search graph, described by us in Section 2.3, could be optionally returned by Moses using -output-search-graph FILE parameter. Resulting text file contains list of hypotheses that were considered during the decoding stage. Each line in the file represent such hypothesis. Contents of file is ordered by sentence identifiers, that appear in the beginning of each line indicating to the sentence that was being translated when the hypothesis formed. The hypotheses are in one of three possible formats.

The *initial hypothesis* refers to the empty hypothesis that is used as starting point in process of hypothesis expansion. This hypotheses have a simple structure and don't carry any additional information:

```
0 hyp=0 stack=0
```

The *regular hypothesis* encodes information about translation decisions that were made by the decoder in a given state. Sample regular hypothesis has following format:

```
0 hyp=10 stack=1 back=0 score=-1.497 transition=-1.497 forward=3437
fscore=-8.900 covered=0-0 out=del Parlamento
```

The line starts with sentence identifier which is followed by a unique hypothesis identifier hyp=10. This identifier is used to reference hypotheses. The stack in which hypothesis is placed identified by stack=1, this is also the number of words covered in original input. The next attribute back=0 is a reference to the previous hypothesis,

in this case it refers to the initial empty hypothesis. This means that current hypothesis represents the first translation option addition. Overall score of current partial translation is expressed as score=-1.497, which derives from a log-probability which is calculated given machine translation models. The cost of transition to current state is available as transition=-1.497. After finalising the translation, hypothesis attributes are enriched by adding information about the best forward step forward=3437 and score fscore=-8.900 to the end of the graph. Original sentence coverage information is available in following format covered=0-0, where 0-0 indicates covered interval. Finally, the last attribute out=del Parlamento contains the translation option that was added by introducing current hypothesis.

The third type of hypothesis groups *recombined hypotheses*. That are omitted as a result of recombination process that was reviewed by us in Section 2.3.5. Recombined hypotheses have following format

```
0 hyp=734 stack=2 back=24 score=-2.684 transition=-1.226
recombined=731 forward=8037 fscore=-7.962 covered=3-3 out=de Apoyo
```

Here additional attribute recombined=731 indicates to the hypothesis with a better score, with which current hypothesis was recombined.

Another option fetching the information generated during decoding is executing Moses with -verbose 3 parameter. This way detailed logs will be provided by the system. The resulting output will contain mostly discarded hypotheses and will not have information about best path that is being added in case of the switch we described earlier. That is why we prefer using the -output-search-graph option instead.

More details are available at [http://goo.gl/6RYbDK]

3.2.2   Preprocessing data for paraphrasing

For initial testing purposes we used Moses to translate set of 3000 Russian sentences from news domain (*newstest2012b*) into English. The resulting search graph dump file was about 5.7GB. In order to process it efficiently we decided to encode information from this file into a relational database. We developed a small Python script that parsed each line in the search graph file and mapped attributes into a dictionary, which was then stored in an SQLite database. The resulting transformation reduced size of our graph representation to 2.3GB. Moreover, main benefit of having our graph stored in a relational database was being able to run SQL queries against it.

Bibliography

Callison-Burch, C. (2007). Paraphrasing and Translation Doctor of Philosophy School of Informatics University of Edinburgh.

Koehn, P. (2009). *Statistical Machine Translation*.

Koehn, P. (2010). Enabling Monolingual Translators : Post-Editing vs . Options. (June):537–545.

Koehn, P. and Haddow, B. (2009). Interactive Assistance to Human Translators using Statistical Machine Translation Methods.

Macklovitch, E., Lapalme, G., and Gotti, F. (2008). TransSearch : What are translators looking for ?

Bibliography

Callison-Burch, C. (2007). Paraphrasing and Translation Doctor of Philosophy School of Informatics University of Edinburgh.

Koehn, P. (2009). *Statistical Machine Translation*.

Koehn, P. (2010). Enabling Monolingual Translators : Post-Editing vs . Options. (June):537–545.

Koehn, P. and Haddow, B. (2009). Interactive Assistance to Human Translators using Statistical Machine Translation Methods.

Macklovitch, E., Lapalme, G., and Gotti, F. (2008). TransSearch : What are translators looking for ?