Machine Translation for Chinese-Spanish:

Experimenting with online Statistical and Rule-Based Paradigms.

Master Thesis

Author: Jordi Centelles
Advisor: Dr. Marta R. Costa-Jussà

July 11, 2013
Contents

1 Introduction 12
  1.1 Objectives .................................. 13
  1.2 Structure .................................. 13

2 Machine Translation 14
  2.1 Historical Overview of MT ..................... 14
  2.2 Approaches to MT ............................ 15
  2.3 Major issues with statistical and rule-based MT .. 17
  2.4 Pre-editing and post-editing .................... 18
  2.5 Main challenges between Chinese and Spanish .... 18

3 Statistical MT 19
  3.1 Language Model .............................. 21
  3.2 Translation Model ............................ 21
  3.3 Log-linear Model ............................. 22
  3.4 Evaluation .................................. 23

4 System architecture 25
  4.1 Client-Server model .......................... 25
  4.2 Backend .................................... 26
    4.2.1 Server .................................. 27
    4.2.2 Webserver ................................ 28
  4.3 Frontend .................................... 29
    4.3.1 Building the web “www.chispa.me” .......... 29
    4.3.2 Android App .............................. 35
    4.3.3 Iphone App ............................... 41
    4.3.4 Related work and comparison ................. 42

5 Evaluation of the SMT system 44
  5.1 SMT configuration details ...................... 44
  5.2 Data Collection ................................ 44
  5.3 Experimental Results ........................ 45
5.3.1 Automatic Evaluation ............................................. 45
5.3.2 Human Evaluation ................................................ 46

6 Rule-based MT ................................................................ 49
   6.1 Monolingual dictionaries ........................................... 50
   6.2 Bilingual dictionaries and Transfer Rules ...................... 51
   6.3 3-level transfer rules ............................................... 52

7 Contributions and Further work ..................................... 54

Appendices ......................................................................... 56

A Research Publications ...................................................... 57

B Rule-based Toy System ..................................................... 69
   B.1 Chinese monolingual dictionary ................................. 69
   B.2 Spanish monolingual dictionary ................................. 70
   B.3 Bilingual dictionary ................................................. 74
   B.4 Transfer rules ......................................................... 75
List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Vauquois’s pyramide</td>
<td>16</td>
</tr>
<tr>
<td>2.2</td>
<td>Scheme</td>
<td>18</td>
</tr>
<tr>
<td>3.1</td>
<td>Word-based models</td>
<td>19</td>
</tr>
<tr>
<td>3.2</td>
<td>Phrase-based models</td>
<td>20</td>
</tr>
<tr>
<td>3.3</td>
<td>N-gram matches</td>
<td>23</td>
</tr>
<tr>
<td>4.1</td>
<td>Client-server model</td>
<td>25</td>
</tr>
<tr>
<td>4.2</td>
<td>Global structure</td>
<td>26</td>
</tr>
<tr>
<td>4.3</td>
<td><a href="http://www.chispa.me">http://www.chispa.me</a></td>
<td>30</td>
</tr>
<tr>
<td>4.4</td>
<td>Pinyin Input Method</td>
<td>31</td>
</tr>
<tr>
<td>4.5</td>
<td>MYSQL databases</td>
<td>33</td>
</tr>
<tr>
<td>4.6</td>
<td>Spanish suggestion</td>
<td>33</td>
</tr>
<tr>
<td>4.7</td>
<td>Chinese suggestion</td>
<td>34</td>
</tr>
<tr>
<td>4.8</td>
<td>Accessing database</td>
<td>34</td>
</tr>
<tr>
<td>4.9</td>
<td>User’s suggestions displayed in the PHP file</td>
<td>35</td>
</tr>
<tr>
<td>4.10</td>
<td>OCR</td>
<td>36</td>
</tr>
<tr>
<td>4.11</td>
<td>Translator Application</td>
<td>37</td>
</tr>
<tr>
<td>4.12</td>
<td>Landscape mode</td>
<td>38</td>
</tr>
<tr>
<td>4.13</td>
<td>Translation with images displayed</td>
<td>40</td>
</tr>
<tr>
<td>4.14</td>
<td>Also it is possible to introduce text with voice</td>
<td>40</td>
</tr>
<tr>
<td>4.15</td>
<td>Android Market</td>
<td>41</td>
</tr>
<tr>
<td>4.16</td>
<td>Translator app for Iphone</td>
<td>42</td>
</tr>
<tr>
<td>6.1</td>
<td>Analysis-transfer-generation</td>
<td>50</td>
</tr>
</tbody>
</table>
List of Tables

4.1 Pinyin outputs for "wo" ............................... 32
5.1 Training corpus statistics (M and K stand for millions and thousands, respectively) ........................................ 45
5.3 Chinese-Spanish results with BLEU .......................... 46
5.4 Human evaluation ........................................... 47
5.2 Tunning and Test corpus statistics (M and K stand for millions and thousands, respectively) ............................. 48
Resum

A causa de l’imparable creixement de la informació en múltiples idiomes, la traducció automàtica ha esdevingut una aplicació necessària en el nostre dia a dia. Així ho demostren les contínues inversions que empreses tecnològiques duen a terme per desenvolupar nous i millors sistemes de traducció. Parelles de llengües tals com el xinès i el castellà són freqüentment utilitzades tant en els negocis com en la vida quotidiana.

Aquest projecte se centra en la traducció online entre xinès i castellà. Inicialment, es fa una breu introducció al camp de la traducció automàtica, un repàs sobre els principals punts de la seva història i s’expliquen els principals enfocaments de traducció automàtica que hi ha.

A continuació, s’introdueix la traducció automàtica estadística ja que és el paradigma de traductor que hi ha darrere del nostre sistema online. S’explica teòricament quines són les seves bases matemàtiques i a partir de quins models es crea un traductor automàtic estadístic. Així mateix, també s’explica teòricament com funciona un mètode d’avaluació per aquest tipus de traductors que, més endavant en el projecte, serà utilitzat per avaluar el nostre propi traductor.

En la part següent es descriu com està construït el sistema online que ens permet traduir tant des d’una pàgina web com des de dues aplicacions mòbils (una per a sistemes Android i l’altra per a sistemes iOS). Així mateix, s’explica com han estat inclosos uns mètodes especials d’escriptura tant de caràcters xinesos com de caràcters llatins per a la web i les aplicacions.

Posteriorment, presentem la part d’avaluació del sistema de traducció estadístic que ens dóna informació tant de la qualitat de les traduccions com dels corpus lingüístics utilitzats per entrenar el sistema.

Finalment, canviem de paradigma per centrar-nos en la traducció basada en regles també entre xinès i castellà. S’explica de forma teòrica com funciona aquest nou sistema de traducció i es construeix un sistema senzill de mostra per ensenyar com s’estructura aquest altre tipus de traductor. Aquesta última tasca constitueix les bases d’un sistema de traducció per regles de codi obert que es desenvolupà en el context del Google Summer of Code 2013.
Abstract

Due to the overwhelming increment of information in multiple languages, Machine Translation has become an essential application in our lives. Proof of this are the continuous investments made by technology companies to develop new and improved translation systems. However, translation between distant language pairs such as Chinese and Spanish, which are commonly used in both business and daily life, have been seldom addressed from a research point of view.

This project focuses on online translation between Chinese and Spanish. Initially, we present a brief introduction to the field of Machine Translation, and a quick overview to its history and its main approaches.

After that, we introduce Statistical Machine Translation, which is the translation paradigm behind our online system. We explain the mathematical basis and the structure of the models. Likewise, we present the evaluation framework used to evaluate our system.

Next, we describe how the online system has been built, which allows for translating either from a web-based interface or from two mobile applications (one for Android and one for iOS). Also, we explain how special methods for inputting Chinese and Latin characters are included in the web-based interface and the application.

Then, we present the implementation details regarding our statistical translation system. We cover both the description of the corpora used to train the system and the quality assessment of the resulting translations.

Finally, we briefly explore the paradigm of rule-based machine translation (also between Chinese and Spanish). We explain the theory of this translation system and we describe the construction of a toy system in order to illustrate how this kind of systems works. This last task is the basis for an open-source rule-based machine translation system that is being developed within the framework of the Google Summer of Code 2013.
Agraïments

I would like to thank the Institute for Infocomm Research and Baidu for giving me the opportunity to start and develop this great project in Singapore. Also I would like to thank the people of the Google Summer of Code and Apertium for accepting my proposal and letting me continue investigating into the fascinating world of Machine Translation.

A Rafael E. Banchs porque junto con Marta R. Costa-jussà me han dado todo lo que necesitaba para hacer este proyecto: conocimientos, motivación e ilusión. Gracias a ellos mi estancia en Singapur no podría haber ido mejor.

Gràcies per tot Marta, has estat una tutora excepcional.

També a la UPC per tots aquests valuosos anys de la meva vida ensenyant-me dia a dia, no només sobre les telecomunicacions sinó també sobre l’esforç i el treball dur.

A la meva família i a la Mònica pel seu recolzament i confiança en mi i en especial, a la meva mare, pel seu amor incondicional i perquè cada dia ho dóna tot pels seus fills.

Però per sobre de tot, aquest projecte és pel meu pare, que vagi on vagi sempre el porto al cor. Va per tu.
Chapter 1

Introduction

One of the most important characteristics of our society is that the quantity of multilingual information, thanks to the new technologies, is growing incredibly quickly. As our goal is to keep the communication and the knowledge growing, we must get over this handicap that appears when we need to deal with some information written in an unknown language.

Furthermore, every day we are in a more globalized world. Therefore, companies, governments, and most people have to face the challenge of communicating in languages different than their own.

This leads to an interesting point for the translation: For preventing that business and personal communications are being stopped due to the lack of language comprehension, we strongly need powerful and reliable translation tools.

Spanish and Chinese are two of the most spoken languages in the world. An incredible number of business are set every day between their people and due to migration movements, more and more Chinese people are nowadays living in Spanish speaking countries and the other way around. Yet, even when well-known companies are investing lots of money and dedicating its researchers to develop powerful translators, most translation systems among these two languages do not provide acceptable translations. This demonstrates that translation and more precisely, translation between two distant languages is a challenging task.

As mentioned, information and knowledge are being spread much more every day thanks to new technologies. Smartphones are one of these new. Its reduced size makes these devices totally portable and their powerful processors make them able to run many kinds of applications. Therefore, the context of our project offered a great opportunity to develop a translator app that could be useful in every moment we are in a foreign country.
1.1 Objectives

The main objectives of this project are listed as follows:

- Build a user-friendly interface for an online statistical translator between Chinese and Spanish.
- Set up the whole structure needed to connect the user interfaces with the statistical translator.
- Develop Google Play (Android market)\[4\] and “App Store” (iOS market)\[2\] translation mobile applications intended to ease life in either a Chinese or Spanish speaking countries.
- Build an entire rule-based Machine Translation system that translates from Chinese to Spanish.

1.2 Structure

This project is divided into 7 chapters. After this introductory chapter, the following ones describe mainly the two types of MT systems we have been dealing with and the user interface created.

Chapter 2 presents a brief overview of MT including history and its main approaches.

Chapter 3 describes the theory behind the popular phrase-based SMT approach.

Chapter 4 reports a description of the whole structure of the online Chinese-Spanish translation system including the frontend (website and mobile applications).

Chapter 5 shows the experimental details and the translation quality of the SMT system behind our online system.

Chapter 6 reports the rule-based MT system we are currently working on. Also, in order to show how a this system works, a toy system is described.

Chapter 7 summarizes the conclusions and further work
Chapter 2

Machine Translation

Machine Translation (MT) is the use of computers to perform the mapping from one source language to a target language. In turn, MT is commonly considered to be a subfield of computational linguistics[13] which deals with the use of computers to perform natural language processing. MT is becoming every day more and more important due to the need of translations of technical and commercial documentation which is growing much faster than the human capacity of translation.

Imagine this scenario. You have to either set a business with a Chinese Company by videoconference or call a colleague in Hong Kong. Neither him speaks Spanish nor you speak Chinese. However, you are able to have a conversation! You speak in Spanish and a speech translator in the cellphone translates simultaneously what you say into Chinese and you hear your counterpart replies but in Spanish. This scenario is not yet real but thanks to the combination of technologies like ASR[29], MT and Text-to-Speech (TTS)[35] we are getting really close to it.

An MT system essentially takes a text in one language (called source language), and translates it into another language (called the target language).

2.1 Historical Overview of MT

History of MT is huge and full of details. Some experts trace MT back to the era Before Christ, others only back to 1950s. In this section, we are offering a brief overview of it. The text that follows summarizes and refers to ideas extracted directly from the following sources [23][37][26].

Early ideas of MT may be settled back in the seventeenth century with philosophers such as Descartes and Leibniz that proposed codes that would relate words between different languages.

In later centuries, there was a proposal for interlingua languages (for example the popular Esperanto by Lázar Zamenhof 1887). However, there were no attempts to mechanize translation until the mid-1930s. George Artsrouni and Peter Troyanskii
made the first patents for translating “machines”.

The modern history of MT begins in the post WW2 and Cold War era, when the race for information and technology motivated researchers to find a way to quickly translate information: the first proposals for MT using computers came back after the Second World War, in July of 1949, from Warren Weaver, a researcher at the Rockefeller Foundation. Mostly, it was based on Shannon’s information theory.

The first public demonstration of an MT system was in 1954, the Georgetown-IBM experiment. Even though it was a small translation system (with 250 words and translating just 49 carefully selected Russian sentences into English) it became really popular in the American society. Moreover, it showed not only in the United States, but in the world that MT was a powerful tool to consider.

In the 1960’s MT had its own dark age: Automatic Language Processing Advisory Committee (ALPAC) was a commit of seven scientists set by the US Government to assess the development that the MT field was living. It concluded that MT would never reach the quality of human translations, besides it always would be slower, more expensive and less accurate than human translation despite all the investments that the US government had done. It saw no need for further investment in MT research.

That inform caused that research in this field was abandoned for more than a decade. However, in the following decade MT research took place largely outside the US: the investment in Canada, France, Germany, UK and Russia continued. In the US though, some companies like Systran and Logos kept his development in MT even the lack of interest by the US government.

During the 1980s the field was being especially developed in Japan by well-known electronic firms: Toshiba, Mitsubishi, Sharp, Fujitsu, etc. They were focused on building a machine translation system from and to English.

It was in the early 1990s that in the US, MT was starting to be installed in personal computers rather than highly powerful computers.

This field has seen major changes in the last years. One of the most significant development is the appearance of commercial MT systems.

For example, companies like Google, Microsoft, Asia Online, etc. use MT commercially.

### 2.2 Approaches to MT

MT approaches can be classified following different criteria. Whether we use the knowledge representation as a criterion, MT can be divided into three classes following the popular Vauquois pyramid: direct, transfer-based and inter-lingual.
Direct MT is the oldest and the less popular approach. It maps the source language to the target language in a directly way. Direct translation approach needs only a little syntactic and a semantic analysis.

Transfer-based approach first convert source language text into an abstract representation which depends on the source but not the target language. In the second stage, source language oriented representations are converted into target language representations. Finally, this target language representations are converted into the target language.

Inter-lingual approach translates the source language to an intermediate form called interlingua and then from this to target language. Interlingua may be either an artificial one or auxiliary language.

Also we can classify MT following the source of information criterion and in this case there are mainly three types of MT systems: the ones based in linguistic rules, the ones based in data corpus and the hybrids (a mix of the other two).

Rule-based MT is conducted by retrieving information about the semantics, morphology and syntax of the source and the target language from bilingual dictionaries and grammars. Then with this information, the grammar rules are built and with the appropriate software are processed to obtain the best translation of the source sentence. Rules are typically designed by using linguistic knowledge [34].

Corpus-based MT systems typically learn translations from large collections of text. In turn, these approaches can be classified in:

---

1 Image extracted from http://en.wikipedia.org/wiki/Machine_translation
– Statistical MT: The process consists of using probabilities previously calculated in a training stage for the translator with bilingual text corpora, to search the words in the target language that best translates the words in the source language. Yet, after finding them, put the words in the right order is basic so the sentence makes sense in the target language. [15]
– Example-based MT: With the help of a bilingual corpus, it translates by analogy using other translations as examples as long as the syntactic structures are similar and words in the input sentence can be replaced with the words in the example sentence. [24]

• Hybrid MT combines the strengths of rule-based and statistical machine translation. We can say that it merges the predictability and language consistency of rule-based MT with the fluency and flexibility of statistical MT. Approaches integrating more than one MT paradigm are receiving increasing attention. [21]

Basically Hybrid MT has been approached mainly as follows:
– Rule and statistical concatenation: Rule and statistical systems are concatenated in serie.
– Rule and statistical systems combination. Rule and statistical based systems outputs are combined using different criteria.
– Rule and statistical hybrid architectures. Here rule and statistical systems are combined at intermediate steps.

2.3 Major issues with statistical and rule-based MT

In the case of statistical MT the major issues are:

• The size of the corpus should be relatively large in order to build probabilistic models that allow an accurate translation.
• It works significantly worse between languages with different word orders.

On the other hand, with rule-based MT there are the following problems:

• They are likely to result in Literal translations instead of natural translations.
• There should be a large number of rules in order to obtain accurate translations
• Linguistic rules are to be done by a linguistic expert.
2.4 Pre-editing and post-editing

The translation quality of these systems can be improved by pre-editing and post-editing the input. Pre-editing involves modifying the sentence to make the translation easier to the machine. For instance, if our corpus is without accents, a good pre-editing should delete all the accents of the input. Tokenizing, truecasing or lowercasecasing are typical pre-editing techniques. More complex ones include reordering of the source sentence to better match the target sentence or segmenting the source words to better match the target vocabulary.

The output of the machine translation can be post-edited to improve its quality. Detokenizing and recovering capital letters are typical post-processing. Other more complex may include generating morphology.

2.5 Main challenges between Chinese and Spanish

Spanish and Chinese are two distant languages because they differ in many aspects such as phonetic, morphological, grammatical, semantic, pragmatic, gestual. However, not all of them represent a challenge in translation. The grammatical differences between both languages represent an important handicap as well as the semantic aspect.

Chinese has few morphology compared to Spanish. For example, it neither changes the gender and the number of the words. There is no verb conjugation in Chinese. The tense for example is represented in the context (specially in the adverbs like “yesterday”, “tomorrow”). This is a drawback when translating from a Chinese sentence into Spanish due to the lack of grammatical information. Regarding the syntax, the structure of the sentence in Chinese is similar to Spanish: Subject-Verb-Object (SVO). Yet, in most cases the verb is added at the end of the sentence and thus a big reordering is needed to translate fluently from Chinese to Spanish.

Besides there are no articles in Chinese, there is also a problem that needs to be solved in order to obtain correct translations: in Spanish, the pronoun as subject is missing due to redundancy. However, in Chinese all the pronouns must appear.
Chapter 3

Statistical MT

This chapter reports the theory of statistical MT which is the translation paradigm that we use behind our online translator and applications.

One of the first works in Statistical MT were the IBM models[17], which mainly compute lexical translation probabilities. Apart from the lexical translation they take simple reorderings into consideration.

![Word-based models](image)

Figure 3.1: Word-based models\(^1\).

\(^1\)Image extracted from Kohen,P. "Statistical Machine Translation"
However, after the IBM models proposal, various projects showed that translation’s quality improved with the use of phrase translation:

- Words may not be the best atomic units for translation, due to frequently one word translates into more than one word in other language.
- To resolve translation ambiguities it is better to translate groups of words instead of isolated words.
- With a large training corpora we can learn longer phrases, even memorize the translation of an entire sentence.

This helps the decoder to get better results.

![Phrase-based models](image)

**Figure 3.2: Phrase-based models\(^2\).**

In this project we used the phrase-based statistical MT approach with Moses\(^{[27]}\) (statistical MT system licensed under LGPL). The word-based and the phrase-based models are originally based on the noisy-channel model\(^{[12]}\):

Imagine you are given a sentence \(f\) in a foreign language and you want to translate it into an English sentence \(e\). We can model those possible translations with a probability distribution: \(pr(e/f)\). We use Bayes rule to write a standard approach to rewrite the translation probability for translating a foreign sentence \(f\) into English \(e\) as:

\[
pr(e \mid f) = \frac{pr(f \mid e)pr(e)}{pr(f)} \tag{3.1}
\]

Obviously, there is not only one correct translation. An acceptable way to find the best translation from \(f\) to \(e\) is to find the \(e\) which maximizes this conditional probability. Moreover, as \(f\) is fixed, the \(pr(f)\) is equal to 1 then:

\[
\text{argmax}_e pr(e \mid f) = \text{argmax}_e pr(f \mid e)p(e) \tag{3.2}
\]

\(^2\)Image extracted from Kohen,P. "Statistical Machine Translation"
The rigorous implementation would be to, via an exhaustive search, go through all the strings \(e\) that maximizes the probability. However, this would result in a totally inefficient translator. Thereby, the work of an MT decoder is to perform the search efficiently by limiting search space and time keeping a high quality of the translations.

As we can see, the problem of finding the target language that maximizes the conditional probability has been divided in two parts:

1. the language model \(p(e)\)
2. the translation model \(p(f/e)\)

The aim is to use the data set to train the language model and the translation model and then use those models to search for \(e\) maximizing \(p(f/e)p(e)\) in the decoding process.

### 3.1 Language Model

It is an essential component of any SMT system. It measures how likely a sequence of words is in the target language. In other words, it measures the fluency of the output. Obviously, we want an MT system to output not only the correct translation of the foreign words but also to put them together in a fluent sentence. The LM helps the system to find the right word order. The LM is trained only with the target sentence corpus. That is because we care about the fluency of the target language, not the source language. LMs are based on n-grams meaning that they collect statistics of how words are likely to follow each other. For this project, we used the SRILM tool to build a 5-gram LM.

### 3.2 Translation Model

The Translation Model \(P(f/e)\) gives the probability that the meaning expressed in \(f\) is also expressed in \(e\). Moreover, it provides a model of the sentence-aligned source-target training corpus.

Given parallel corpora aligned at the level of sentence, with the implementation of the IBM models as computed by the GIZA++ program, we perform the word alignment in the corpus. GIZA ++ is an open source software that aligns bilingual text at the sentence level in both directions (\(e\) aligned with \(f\) and \(f\) aligned with \(e\)) and trains translation models from the aligned corpora.

Thus, from now on our bilingual corpus is aligned at a level of words. However, as pointed out before, we are not using anymore the word-based models, instead, we are using the phrase-based models. Hence, from this word aligned corpora we must do a phrase extraction. All possible phrase pairs that are consistent with the
word alignments are extracted. Afterwards, this phrases are scored and the phrase translation table is created.

3.3 Log-linear Model

So far, we have described that phrase-based models are based on the noisy channel model. However, this original approach has been changed in the most recent implementations of phrase-based MT systems. It was observed that the weighting of the different models may lead to improvement in translation quality. The Log-linear model[28], different from the noisy-channel model, allows for introducing additional features to score and combine phrases.

\[
p(x) = \exp \sum_{i=1}^{3} \lambda_i h_i(x)
\]  

(3.3)

where:

- \(x\) is a random variable (e,f)
- feature function \(h_1 = \log(\phi)\) (\(\phi\): phrase translation table)
- feature function \(h_2 = \log(d)\) (\(d\): reordering model)
- feature function \(h_3 = \log(p_{LM})\) (\(p_{LM}\): Language model)

The reordering function at the beginning was performed by the decoder with the distance-based reordering model. However, it is only conditioned word reordering based on word distances and nothing else. Hence, it was necessary to create a reordering model. This reordering model is performed by the lexicalized reordering and it builds a table with all the information that the decoder needs to reorder the words within the output sentence.

These models are all built in the first stage: the training. For the training stage we need the bilingual corpora to train all the models. However, the weights given to them are not the best ones in order to obtain the best translation. This is done in the second stage: tuning. In this phase with another bilingual corpora, the significant advantage of the log-linear model over the noisy-channel approach is exploited: we ‘tune’ the relative importance of the models by varying the lambda values. For example, if we decide that in our case the outputs are not fluent enough, we can give a higher weight to the language model. Finally the last phase is: testing. In the testing stage we are going to “measure” how good the output translations of our system are.
3.4 Evaluation

Mainly, there are two ways to evaluate our machine translation system:

1. By asking bilingual human evaluators to judge the quality of the translations
2. By using an automatic evaluation system that compares the system’s output against one or more human translations of the same sentence (reference translations).

While the first option is much more accurate it is also much more expensive. Therefore, it is more convenient to find an automatic machine translation evaluation system that can score our system. Currently, one of the most popular (and controversial) evaluation measure is the BLEU (Bilingual Evaluation Understudy). The main idea of BLEU[32] is ”the closer a translation output is to a professional human translation, the better it is”.

The score that BLEU computes is based on the number of n-gram matches with the reference translation as shown in the below figure:

So far we have described that BLEU scores a sentence by comparing it with a good quality reference translation. Yet, we are not interested in evaluating the output for a single sentence, our aim is to evaluate the output of our translator for a whole corpus in order to obtain more accurate evaluations. Regarding this, what we perform is having our system already trained and tuned and using a small corpus that is correctly translated, we translate this corpus with our system and evaluate the output comparing it to the reliable translation.

Furthermore, it is important to point out that the corpus used to evaluate the system must not be the same corpus we have used to train or tune it. Because our aim is to analyze how our system is able to translate sentences that it has not been trained with. In other words, we want to see how our system can extrapolate its learning to new inputs.

---

Image extracted from Kohen, P. ”Statistical Machine Translation”
One of the main innovations of BLEU is that score can accept one or multiple reference translations. This is a worthwhile feature because there are multiple ways to translate a single sentence. In fact, this is one of the major shortcomings of the automatic MT evaluation measures. Whether the punctuation is low, this does not mean that our translation is bad but our translation is different than the given translation reference. The more reference translations per sentence there are, the higher the chances of getting a higher score are.
Chapter 4

System architecture

This chapter describes the user-interface and the system architecture of our Chinese-Spanish online web translator and our app mobile applications. The structure of our system is based on the client-server architecture. The client side includes the User Interface which the user uses to interact with the Server side.

4.1 Client-Server model

There are two roles in the network, the client and the server. The client sends a request and the server returns a response. Normally, the server is being shared by multiple client programs and receives multiple requests.

Figure 4.1: Client-server model\(^1\).

This model has become one of the main ideas of network computing. For instance, the Internet’s main protocol, TCP/IP, uses this architecture. Thereby, a

web browser is a client program that requests services (e.g. to receive a Web page) from a Web server installed in another computer via the protocol HTTP (Hyper-Text Transfer Protocol). Also, with the TCP protocol, the client can make requests to a server for files stored in a database that is connected with that server. It is important to point out that in order for a connection to succeed the client has to send the request via the same port that the server is listening.

In our case, we needed to use a server with at least 4 GB of RAM[11] due to all three models (language model, translation model and reordering model) should be loaded in memory in order to obtain the translations in a reasonable time.

In the picture below there is represented the entire process. The structure of the process will be explained from right to left, starting with the backend (server side) and ending with the frontend (the user’s interface):

![Figure 4.2: Global structure](image)

### 4.2 Backend

The backend is divided by the server where Moses and the pre-processing are running and the Webserver, the part of the server that interacts with the User-Interface.
4.2.1 Server

As pointed out before, we will use for decoding the Moses software. Nevertheless, every time Moses is started it has to load all their models and this takes about 30 seconds (depending on the size of the models). Therefore, we need Moses to be permanently running in “server mode”. For this, we will use the mosesserver.pl script (written by Herve Saint-Amand[6]). This script is listening on one port and every sentence that goes through this port is sent by the script to the moses program in order to be translated.

Apart from this, the main function of this script when is executed is to load the moses software along with the moses.ini file where all the properties (e.g. translation direction) are specified. Moreover, it is in charge not only to send the query received from the PHP files to the moses software, but also to send it back to the appropriate PHP file.

Before that, the sentence sent by the user, passes through a pre-process. It consists of four scripts: Three for Spanish and one for Chinese. In the case of the Spanish:

1. The first script is called unaccent and it is embed inside the PHP file. Its function is to remove all the accents entered by the user. That’s because, removing all the accents in the pre-process gets better results than to keep them with the possibility that the user makes spelling mistakes with the accents and this causes wrong translations. The important point here is that the accents are only removed from the Spanish-to-Chinese corpus. Yet, in the other corpus the accents are kept. Thereby, when we enter a Chinese character that its translation in Spanish is an accented word, the translator will find the accented word in the corpus and will output it.

   "La traducción automática, también llamada MT" =⇒ "La traduccion automatica, tambien llamada MT"

2. The second step is the lowercase.pl provided by Moses. Its function is to change all the capital letters to small ones.

   "La traduccion automatica, tambien llamada MT" =⇒ "la traduccion automatica, tambien llamada mt"

3. The tokenizer.pl also provided by the moses program, breaks the stream of text up into words. That means separate the words between them and also between symbols (commas, dots, question marks etc.)

   "la traduccion automatica , tambien llamada mt" =⇒ "la traduccion automatica , tambien llamada mt"

This makes much more effective the translation because, if in the input sentence there is something like “wordA,wordB”, the moses will take this as a unique word and then the translation will be wrong. Therefore, the function of the tokenizer is to split this: “wordA , wordB” so the moses will recognize two words.
As these two perl programs do not need to load anything when they are being
started, we realized that they are open fast enough to not have to implement any
server mode script for them. Then, every time a sentence is introduced and sent to
the PHP, the programs are executed and closed until the next sentence is received.

On the other hand, we have the Chinese preprocessing. For this task, I mixed up
two open-source scripts available on Internet. One of them is the Stanford Chinese
Segmenter[9]. This script in perl is created for splitting a Chinese sentence not into
symbols but into words.

One of the features of the Chinese language is that one Chinese word can be com-
posed of two Chinese symbols. For instance, the word “contradiction” in Chinese
is composed of these two symbols: 矛盾. Furthermore, these two separate symbols
have their own meaning: “矛” means lance and “盾” means shield.

Another feature of this Asian language is that in a sentence all the words are
written together. This is the other reason why we need the Stanford Segmenter.
Because a sentence must be splitted into words before entering to the translator.
Here you have an example of how the segmenter works: 小明和小在花里面 → 小明
和 小 在 花 里面

Since each time the Segmenter starts has to load dictionaries (and it takes more
than a half minute) we need to keep it running as server mode like Moses. That
is the second script’s function (stanford-Chinese-segmenter-server by Ruey-Cheng).
It maintains the segmenter listening on one port where all the Chinese queries are
sent by the PHP.

When the sentence is already segmented it is sent back to the same PHP file.
After that, the PHP file sends it through the port that the mosesscript for Chinese
to Spanish is listening and it gets translated.

4.2.2 Webserver

There are four PHP files on the web server. Two of them communicate with the web
and the other two with the mobile applications. In both cases, one of the two PHP
files is related with the Chinese to Spanish translations and the other one with the
Spanish to Chinese. The functions of the PHP files related with Chinese to Spanish
translations are:

1. Receive the Chinese queries from the frontend.
2. Execute the Chinese preprocessing script (Stanford Segmenter) with the Chi-
inese queries as parameters.
3. Send these preprocessed queries to the Chinese to Spanish mosesscript.
4. Receive the queries translated.
5. Send them back to the frontend.
And on the other hand, the functions of the PHP files related with Spanish to Chinese translations are:

1. Receive the Spanish queries from the frontend
2. Execute the Spanish preprocessing scripts (unaccent, truecaser, tokenizer) with the Spanish queries as parameters.
3. Send these preprocessed queries to the Spanish to Chinese mosesscript
4. Receive the queries translated.
5. Send them back to the frontend

4.3 Frontend

4.3.1 Building the web “www.chispa.me”

The aim was to design and build an easy and clear website with the latest technologies in web design. A webpage without advertisements, easy to use, with the minimum number of buttons and effective. The structure of the web consists in HTML language and Javascript code embedded on it.

Hypertext Markup Language[20] is the most widely used language for writing web-page code. It is called a markup language due to its use of tags enclosed in angle brackets (<html>) that allows us to distinguish layout and style from the text. The browser reads the html file and displays the text in a way that the web programmer had previously specified with the tags.

The other main language is Javascript[16]. Its main characteristics are that it is an interpreted language and it is mainly client-side. Interpreted means that executes the source code directly, without a previous compiling process. Client-side is because, even the web page script is stored in the web-server, the Javascript code is executed by the browser.

The Javascript code inside HTML starts with this tag:

```
<script type="text/javascript">
```

and ends with `</script>`. Inside, there is written the code that has to be interpreted by the client’s browser.

Html was used to build the structure of the web and its appearance and Javascript code was used to program the client-side scripts that let the client interact with the web. The Pinyin Input method and the Spanish keyboard are examples of Javascript code.

One of the latest programming technologies that I used to build the website was Ajax (Asynchronous JavaScript And XML)[1] and specifically, jQuery. Ajax allows you to send information to the server and receive information from it without having to refresh the page. This feature gives the appearance of a more current and modern website rather than having the page refreshed every time that a sentence is sent to be translated.
Ajax is as Javascript, a client-side technology. The main point is that the connection with the server is running in background meanwhile you are using the website and this allows the asynchronous send of information between the browser and the client.

![Chinese-Spanish Translator](http://www.chispa.me)

In 2006 JQuery, a new Javascript library was created by John Resig and developed by the jQuery team. Its purpose is to simplify the client-side scripting of HTML and the developing of ajax scripts inside Javascript code. The jQuery library is called in the head of the html document with `<script type = "text/javascript" src="jquery.js"></script>`. The jQuery library contains all the Ajax functions.

In order to have a user-friendly website, we tried to avoid having buttons. Thus, to remove the “translate button”, we add in the javascript code a timeout: Each time the user stops typing, the timeout starts. After 0.5 seconds, the ajax function is executed and the query is sent to the server. However, since the user might introduce a sentence with the copy paste function, the javascript also foresees this: the query is sent when the control key along with the “v” key are pressed, also after 0.5 seconds.

**Text Inputs**

One of the issues of using Chinese and Spanish as a pair of languages was the keyboard. Usually neither Spanish people can write Chinese characters nor Chinese people can write Spanish characters because of their keyboard. We had to solve this handicap. Therefore, we had to include in the website two input methods. One for Spanish and one for Chinese.

For the Spanish I used a virtual keyboard. Its code was already written in Javascript and it’s open source. The code was slightly modified to adapt it to the
website. The text area in the HTML code where the Spanish text is displayed is defined as class = "keyboardInput". Hence, every time a user clicks on any letter of the keyboard, that letter is sent directly to the Spanish text area. It is important to point out that the keyboard.js, the keyboard.css (the cascading style sheet where the layout and style of the keyboard are defined) must be declared in the header of the main HTML file.

```html
<script type="text/javascript" src="keyboard.js" charset="UTF-8"></script>
<link rel="stylesheet" type="text/css" href="keyboard.css">  
```

The JavaScript code of the keyboard was kept apart from the website file due to maintain clarity.

On the other hand, we had to include also a Chinese input method. The first idea was to adapt a Chinese keyboard to the web, but it was not possible owing to the huge number of Chinese characters. There weren’t worthwhile Chinese keyboards available. The solution we found was to adapt a pinyin input method. It is based on the pinyin method of romanization. The Pinyin is the system to transcribe Chinese characters into Latin script: “the Pinyin method allows a user to output Chinese characters by entering the pinyin of a Chinese character and then presenting the user with a list of possible characters with that pronunciation”.

![Figure 4.4: Pinyin Input Method](image)

In this case, with the sound “wo” we might choose among ten tones and each of them represents one Chinese symbol which its meaning. The first six are:

1. 我
2. 窩
3. 臥
4. 顢
5. 萊
6. 蘲
The code is written in HTML and Javascript. There was also a css file defining its style. After reading and understanding how the code worked, we adapted into the web in two steps:

1. We cleaned the code. That means all the useless parts for my website were deleted or modified.

2. Our intention was to keep the pinyin input method hidden unless the user clicked a link. Then, the pinyin will be displayed on the web.

To achieve this, was necessary to use the load method. When the load method is called, it displays the html file that you passed as a parameter. However, this html file cannot has embedded in it any type of javascript code. Thus, the pinyin code had to be divided into two parts: the javascript part and the html part called pinyin.html. The javascript part was embedded into my web page code and the html and the css files were kept apart. As we did with the keyboard, the css had to be defined in the head of the website html file: 

```
<link rel="stylesheet" type="text/css" href="pinyin.css">
```

The pinyin.html was passed as a parameter of the load function and is stored at the same web server folder as the other files:

```
$("#button").click(function() $("#divAbout").load("pinyin.html"); );
```

When the button “button” is clicked, the pinyin.html will be displayed inside the div (The <div> tag defines a division or a section in an HTML document) identified by the id “divAbout”.

### Translation Database

One of the options for helping the translator improve is to include an option in the website that let users suggest a better translation that the one they obtained. When the user thinks that the translation in the target language could be better, he can send a improvement. Therefore, a database that can store all these suggestions had to be included in the process. To create the databases I used an open source database management system: MySQL[8].

Using the UNIX console of the server I created a database called “Translation” and then inside of it two tables: “correctionsES” (where the corrections being
Spanish the target language are stored) and “correctionsZH”. Each of the tables have three columns: Es,Zh and Corrections. Where inside Es column there are the Spanish sentences entered or received by the user, in Zh the Chinese ones and in corrections, the suggested translations. It is easy to know whether it is an improvement on a Spanish translation or a Chinese translation because in the website there are two inputs: one for the Spanish suggestions that stores them in the “correctionsES” table and the other one for improvement in Chinese that go in the “correctionsZH” table.

```
mysql> show tables;
+---------------------+
| Tables_in_translator |
| correctionsES       |
| correctionsZH       |
| users               |
+---------------------+
3 rows in set (0.00 sec)
```

Figure 4.5: MYSQL databases

Figure 4.6: Spanish suggestion
Moreover, I created another table inside translations, called users. In this table there is only one username and one password. The username and the password are required to access into the PHP. The aim is to control who can access to this information.

The PHP file shows the suggestions in a more clear way compared to the console.
Another important point to improve, not the quality of the translations but the speed, is to include in the process a data file where all the translations done are written in it. The reason of this is because when a huge number of translations have been done, we can do statistics and know which are the most common queries entered by the user. Then, the idea is to store these translations in a database and when someone enters one of them in the web, instead of sending it to the translator, it will provide the result directly from the database. Thus, the speed is increased and the translator is less saturated (it will be further implemented if we see that the web receives lots of petitions). Given that the translations are stored automatically in the data file, it was necessary to implement another timeout. However, in this case the timeout is much bigger than the previous ones. This is because we only want to store in the data file the entire sentence, rather than chunks of the sentence. Hence, after 3 seconds since the user left the keyboard, the sentence is sent to the data file.

4.3.2 Android App

The android app was programmed with the android development tools (ADT). It is a plugin for the Eclipse IDE that gives you the necessary environment for building an app with the programming language Java. At the beginning the idea was to build a simple translator app only. Yet, later some extra features were added to the app in order to make the app not only a translator but rather a useful tool when you are in a foreign country. Also, with the app we had the same problem as with the
website. How a Spanish person could enter to the translator some Chinese words he is observing, for example, in a Chinese signboard? We realize that the appropriate feature the app must include is an OCR (Optical Character recognition)[33]. An OCR is the electronic conversion of scanned images into machine-encoded text.

Therefore, we had to find an open-source OCR to include in my app. Tesseract[10] is an open-source OCR, released under the apache licence. It is considered one of the most accurate free software OCR engines available nowadays.

Besides, there is also a simplified project of Tesseract called Tess-two (developed by Robert Theiss) that contains an Eclipse Android library that provides a Java API. You need to build the project Tess-Two and import the project as a library in Eclipse. Furthermore, there was an OCR app developed by Robert Theiss that we used for our project: it performs optical character recognition on images captured using the device camera.

Embed the OCR inside the app was not an easy task: one of the main issues was how to achieve communication between the OCR and the app. That is, when the user had made a photo and the OCR had recognized the text, how can we get that this text is sent to the translator app? Well, the answer was using the android “intent” method. Intent allows you to open a secondary app inside the main app and when the secondary is done, it is closed and it sends the result to the main app with the “putExtra” method.

Another basic point was to send the sentence entered by the user from the app to the translator. For this purpose, as with the website, there is stored one PHP file in the web-server waiting for the sentences. Thereby, the communication with the PHP file is performed with the “HttpClient” Interface. Among other things, it
allows a client to send data to the server via for instance, the POST method as we used on the web.

After creating a new HttpClient, you specify the url of the server, the data that has to be sent and its codification (it is important to specify a UTF-8 codification due to the Chinese characters). Finally with HttpEntity you settle down that the app has to be kept waiting for a server’s reply. When the translation is received, the handler method is in charge of display it in the appropriate textfield.

Figure 4.11: Translator Application

Furthermore, it is important to include a conversion function from a Stream of bits (it is what we receive from the PHP file) to a String (what we display).

We needed to define two HttpClient functions: One for the translations from Spanish to Chinese (the sentences are sent to the application-es-zh.php file stored on the webserver) and another for the other way around (application-zh-es.php also in the webserver). We have also implemented a very simple but effective language detection system, which is very suitable for distinguishing between Chinese and Spanish. Given the type of encoding we are using (UTF-8), codes for most characters used in Spanish are in the range from u0000 to u0080. Accordingly, if all the characters of the sentence entered are inside this range, the sentence is automatically sent to the es-zh.php. Otherwise, they are Chinese characters and they are sent to zh-es.php. Once the translate button is pressed the sentence to translate is sent to the PHP file via the HttpClient method.

There was another function that, it may not be essential, but we thought it could facilitate user’s work. It is the copy button. When it is pressed it copies the output
of the translation on the clipboard. This means that you can paste the translation in every other text field (for instance, you may open your smartphone’s browser and paste in a google search field the Chinese sentence you already translated and thus, obtain the search results.)

Another feature is the fullscreen class. It allows the user to show in fullscreen mode the translation of his sentence. It can be useful if for example, you are in a taxi in China and you want to go elsewhere. You can write “Quiero ir a la calle X”. The sentence is translated and if you press the fullscreen button, the Chinese sentence will be displayed in the whole screen which is easier to show it to the taxi driver.

As mentioned before, our goal was not only to build a translator app but an app that would be useful when you are in a foreign country. Thus, some extra tools were added. Imagine this scenario: you are reading a Chinese menu and you want to translate some dishes. However, even you have the translation in your language you realize that a picture would be much more useful than the simple translation. This is the reason why we implemented a image retrieval system that interacts with the flickr servers and provides you several pictures with images of what you had translated. The following procedure is implemented: First when the ”search Image” button is pressed we send an URL (using the JAVA HttpClient method) to a flickr server with all the needed information to receive the needed metadata. In the URL we specify the tag (i.e. the topic of the images we want), the number of images, the secret key (needed to interact with flickr) and also the type of object we expect (in our case, a JSON object[7]). When the response of the server is received, using a script we parse the JSON object.

Metadata received when for that tag there are no available images:

![Image of Chinese characters](image-url)

Figure 4.12: Landscape mode
There are two main fields, photos and stat. Stat only reports whether the url sent was correct ("ok") or not ("fail"). However, inside the photos array we find information for every one of the 5 images we requested. In order to avoid the app of crashing when there are no images for a tag, we had included a "if" sentence that closes the thread and displays this sentence on the screen: "No available images"

Afterwards, with the HttpConnection method and the information parsed, we send the URL again to the server and we retrieve the images requested in the app. It is worth to mention that the Java class that implements all these methods extends an AsyncTask in order to not block the UI (user interface) meanwhile is exchanging information with the flickr servers.

The row of images are displayed with an horizontal scrollview in the UI.
The app was uploaded and published on Android Market the 20th of June 2013. Figure 4.15 shows a couple of screenshots of the app:
4.3.3 Iphone App

For the IPhone app we used the xcode software provided by apple and the programming language is called Objective C.[25]

Mainly we have two types of files: the view-controller and the AppDelegate.

For each view in an app, there is created a specialized class for managing views: UIViewController. The view controller will contain all the IBOutlets and IBAction methods for the view. They are all defined in the ViewController.h and in the ViewController.m. Also in the ViewController.xib we can create and modify the view of the app.

The AppDelegate files (AppDelegate.h and AppDelegate.m) are responsible for the life-cycle of the app, the “program controller”. They handle bigger events for instance the startup and shutdown of the app.

The main IBAction in the ViewController.m file is the buttonClick: when the translate button is pressed, the query entered by the user is stored in the variable *username. Then, we specify the method (in our case POST), the codification (UTF-8 because we are dealing with Chinese characters), and depending on which value of the selector the user pressed, we send the query to app-es-ch.php or to app-ch-es.php. After this, the PHP will send the query to the translator as in the other
cases. Then, with connectionDidFinishLoading method we receive the translated query and it is showed on the app screen.

![Figure 4.16: Translator app for Iphone](image)

4.3.4 Related work and comparison

Regarding online translation between Chinese and Spanish we can find some trustworthy translation systems online. Google Translator, which is one of the most used, is based on a statistical paradigm. Google has access to a huge amount of bilingual corpus, that is one of the reasons why translations are quite reliable (specially between those pair of languages involving English). Google translator translates more than 70 languages. For same pairs of languages, Google provides pivot translation instead of direct translation (for example, for Chinese-to-Spanish, they concatenate Chinese-to-English and English-to-Spanish translations). Other online translation systems frequently used are: Microsoft Bing, based on hybrid MT, and Altavista Babylon, based on rule-based MT.

Our online translation system for Chinese-Spanish has the following particularities:

1. It uses a direct translation between Chinese and Spanish, rather than using a pivot language as intermediate step. We have chosen this option because we have enough parallel corpora Chinese-Spanish to achieve a reasonable translation quality.\(^2\)

2. Its user-interface includes the Pinyin input method on the website as a straightforward option to enter Chinese characters.

\(^2\)Also, we do not have access to Chinese-English and English-Spanish data that allows to build a pivot system that outperforms the direct system.
Regarding the translation applications for smartphones, the most downloaded in the Android Market are Google Translator (free) and Talking Translator (free) and in the AppStore are Universal Translator (free) and Translator+(4,99$).

Our application has the following features that most of the translation apps do not include in order to provide a friendly user experience. It combines the base translation technology with:

1. Automatic Speech Recognition (ASR),
2. Optical Character Recognition (OCR),
3. Image retrieval
4. Language detection.
Chapter 5

Evaluation of the SMT system

This chapter reports the experimental details of the statistical MT system behind our web translator and mobile applications [14].

5.1 SMT configuration details

As mentioned before, in order to build our machine translation system, we have used a standard phrase-based SMT based on Moses. This is a corpus-based approach where the key for translation quality is regarding the quality and quantity of the corpus used for training. If corpora from different domains are available it is also relevant how they are combined and which dataset was used to optimize the system. Regarding data preprocessing we have done the following:

1. For Chinese, we have segmented the data using the Stanford Segmenter tool.

2. For Spanish, we have lowercased the data and tokenized it with Moses tools. This preprocessing is kept the same as explained in the previous chapter, but for evaluation purposes.

Moses was used with the standard configuration. Different training domain corpora where concatenated to a single training corpus.

5.2 Data Collection

We have corpora from different domains available. In particular we have the following ones:

- TAUS (both Chinese-Spanish and Chinese-Portuguese). Data provided by this organization include translation memories of technical content.

- EPPS (Chinese-Spanish). This is transcriptions from the European Plenary Parliament Speeches.
• UN (Chinese-Spanish). This is a collection of the resolutions of the General Assembly of the United Nations from Volume 1 of GA regular sessions 55-62.

• BIBLE (Chinese-Spanish). This corresponds to the Holy Bible texts.

• IN-house (Chinese-Spanish). This corresponds to a small corpus in the transportation and hospitality domains.

Corpus statistics for the training corpus can be found in the following tables (table 5.1 and table 5.2):

<table>
<thead>
<tr>
<th></th>
<th>Chinese</th>
<th>Spanish</th>
</tr>
</thead>
<tbody>
<tr>
<td>Taus</td>
<td>Num Sent. (M)</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Words(M)</td>
<td>57</td>
</tr>
<tr>
<td></td>
<td>Vocab(K)</td>
<td>648</td>
</tr>
<tr>
<td>EPPS</td>
<td>Num Sent.(K)</td>
<td>211</td>
</tr>
<tr>
<td></td>
<td>Words(M)</td>
<td>4,3</td>
</tr>
<tr>
<td></td>
<td>Vocab(K)</td>
<td>49</td>
</tr>
<tr>
<td>UN</td>
<td>Num Sent.(K)</td>
<td>58</td>
</tr>
<tr>
<td></td>
<td>Words(M)</td>
<td>1,7</td>
</tr>
<tr>
<td></td>
<td>Vocab(K)</td>
<td>17</td>
</tr>
<tr>
<td>Bible</td>
<td>Num Sent.(K)</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>Words(K)</td>
<td>703</td>
</tr>
<tr>
<td></td>
<td>Vocab(K)</td>
<td>28</td>
</tr>
</tbody>
</table>

Table 5.1: Training corpus statistics (M and K stand for millions and thousands, respectively)

5.3 Experimental Results

Finally, we report the automatic and human evaluation results for Chinese-Spanish.

5.3.1 Automatic Evaluation

We have concatenated all training corpora. For tuning, we have tuned the parameters first with the IN-HOUSE development dataset and, then, with each of the different domains. Results are shown in terms of the standard metric BLEU in the following tables.
Table 5.3: Chinese-Spanish results with BLEU

<table>
<thead>
<tr>
<th>Domain</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chinese-to-Spanish</td>
<td>27.07</td>
</tr>
<tr>
<td>Spanish-to-Chinese</td>
<td>24.22</td>
</tr>
</tbody>
</table>

Percentage scores are up to 100 (the higher, the better) and in this case both are over 20, which means that our translation system outputs readable translations (as a rule of thumb it is considered that any punctuation over 20 points BLEU is quite readable and understandable).

5.3.2 Human Evaluation

We report human evaluation for the Chinese-Spanish system optimized and test with the IN-HOUSE dataset. We were interested in knowing the quality of the system in a touristic domain. The native speaking evaluator was provided with the reference text and the output of the best translation systems. The evaluator was asked to assess the quality of 100 lines from the test corpus using the following fluency scale to measure the acceptability/post-editing effort of the output on segment (sentence) level:

- Flawless (5): refers to a perfectly flowing text with no errors;
- Good (4): refers to a smoothly flowing text even when a number of minor errors are present;
- Non-native (3): refers to a text which contains errors that a native speaker would easily notice and might in some cases disrupt the sentence flow;
- Disfluent (2): refers to a text that is poorly written and difficult to understand;
- Incomprehensible (1): refers to a very poorly written text that is impossible to understand.

Also the evaluator was asked to use the following adequacy scale:

- Everything (5): all the meaning in the source is contained in the translation, no more, no less;
- Most (4): Almost all the meaning in the source is contained in the translation;
- Around half (3): About half the meaning in the source is contained in the translation;
- Little (2): Fragments of the meaning in the source are contained in the translation;
• None (1): None of the meaning in the source is contained in the translation.

Table 5.4 shows evaluation results. We see that punctuations are quite high. Therefore, we are offering a useful translation in the touristic domain.

<table>
<thead>
<tr>
<th></th>
<th>Fluency</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chinese-to-Spanish</td>
<td>4.01</td>
<td>3.91</td>
</tr>
<tr>
<td>Spanish-to-Chinese</td>
<td>3.17</td>
<td>3.01</td>
</tr>
</tbody>
</table>

Table 5.4: Human evaluation
<table>
<thead>
<tr>
<th></th>
<th>Tun</th>
<th>Test</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Taus</td>
<td></td>
<td></td>
<td>Chinese</td>
<td>Spanish</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1394</td>
<td>1394</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>20</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>6.1</td>
<td>5.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1394</td>
<td>1394</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>22</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>6.7</td>
<td>5.3</td>
</tr>
<tr>
<td>EPPS</td>
<td></td>
<td></td>
<td>962</td>
<td>962</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>16</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3.9</td>
<td>5.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1039</td>
<td>1039</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>18</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4.2</td>
<td>5.5</td>
</tr>
<tr>
<td>UN</td>
<td></td>
<td></td>
<td>990</td>
<td>990</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>33</td>
<td>43</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4.3</td>
<td>5.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1010</td>
<td>1010</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>34</td>
<td>44</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4.3</td>
<td>5.5</td>
</tr>
<tr>
<td>Bible</td>
<td>Dev</td>
<td></td>
<td>1033</td>
<td>1033</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>25</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4.4</td>
<td>4.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1035</td>
<td>1035</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>24</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4.3</td>
<td>4.6</td>
</tr>
<tr>
<td>IN-HOUSE</td>
<td>Dev</td>
<td></td>
<td>714</td>
<td>714</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4.1</td>
<td>4.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>751</td>
<td>886</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4.1</td>
<td>4.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>737</td>
<td>894</td>
</tr>
</tbody>
</table>

Table 5.2: Tunning and Test corpus statistics (M and K stand for millions and thousands, respectively)
Chapter 6

Rule-based MT

In March 2013 we started a project proposal for the Google Summer of Code (GSOC) via the Spanish organization Apertium, which is an open source rule-based MT platform.

Our idea was to build an open source rule-based MT system that translates from Chinese to Spanish within the Apertium framework. We realized that it would be a great idea to deal with a rule-based MT system in order to learn how a translator can operate with linguistic rules and dictionaries rather than probabilistic models as it happens in the statistical MT system. After being selected for the GSOC in May 2013, we started developing the system. The project will finish the 19th of September. The rule-based MT system will be freely available in (http://www.apertium.org)

A rule-based MT system consists mainly of three parts:

1. Analysis
2. Transfer
3. Generation

On the analysis part, the input text in the source language is first analyzed morphologically and syntactically. The transfer part is in charge of converting this representation in the source language to another representation but in the target language. Finally, the generation process converts the representation in the target language into a valid sentence in that language. In Figure 6.1 the path of a rule-based system is represented.
In order to build an approach to a rule-based system we have used the Apertium shallow-transfer machine translation platform. Shallow-transfer means that the sentences are analyzed in groups of lexical units rather than parse trees.

Apertium provides an engine and a toolbox necessary to build a machine translation system. Therefore what we had to do in order to complete the translator is to write both monolingual dictionaries (Spanish and Chinese), the lexical dictionary and the transfer rules (to handle the reordering and grammar). The language used to build them is XML[18]. XML is an extensible Markup Language designed to transport and store data. Due to the high complexity in building a rule-based systems we have implemented only one direction: Chinese to Spanish.

In order to show how this type of translator works and for the sake of clarity, the following part exposes and explains the structure of “toy” rule-based system which translates only this Chinese sentence 見小明是一小男 孩 into Jaime es un niño pequeño”. As pointed out before, Chinese is a language where all the words in a sentence are written together, without spaces. This requires the use of a Chinese segmenter like in the statistical machine translation system. Once the sentence has been segmented into words, we can start with the analysis part: 見小明 是 一小 男 孩

6.1 Monolingual dictionaries

In our case we use the Chinese monolingual dictionary (apertium-zh-es.zh.dix) for the analysis part and the Spanish one (apertium-zh-es.es.dix) for the generation part.

The Chinese and the Spanish monolingual dictionaries are presented in the Appendix.

As we can see from the Spanish dictionary, all the tags used in that dictionary
are defined at the beginning. Between the pardefs tags we define the paradigms. A paradigm represents how a particular word inflects in number or gender. In this case we must define 4 paradigms. (Jaime, ser, uno and pequeño)

However, in this toy system we will define the paradigm of the word abstracto that inflects exactly as pequeño in order to show how a certain word can be related to another word’s paradigm:

\[
\langle e\,\text{lm}="\text{pequeño}"\rangle\langle i\rangle\text{pequeñ}/i\rangle\langle\text{par}\,\text{n}=\text{abstract/o_{adj}}/\rangle/e\rangle.
\]

Here we say that the lemma pequeño with root pequeño has the same paradigm as abstracto.

The command below executes the dictionary-based morphological tagging and the simple script called gawk[3] to eliminate non useful parts of the output:

```
echo "小明是一小男孩" | lt-proc zh-es.automorf.bin | gawk 'BEGIN{RS=""; FS=""/";}{nf=split($1,COMPONENTS,"∧"); for(i = 1; i<nf; i++) printf COMPONENTS[i]; if($2 != "") printf("∧%s\$,2);}'}
```

As a result, the following example of analysis is obtained:

小明＜np＞$是＜vblex＞$－＜det＞$小＜adj＞$男孩＜n＞$

### 6.2 Bilingual dictionaries and Transfer Rules

Once we have explained the analysis and generation part, it is time to explain the transfer part. This part consists of two files: the lexical (or bilingual) dictionary and the transfer rules. Once we have the source sentence morphologically analyzed we must translate it into the target language. The bilingual dictionary has the translations of the words and the transfer rules file controls the reordering and the grammar issues. For instance, we want to translate from English “blue cars” to Spanish “coches azules”. The bilingual dictionary will translate word by word (blue→azul, cars→coches). Yet without the transfer rules the output would not be correct: “azul coches”. We need the transfer rules to take care about the reordering (coches azul) and also about the inflections. Whether “coches” is plural we need also blue to be plural: “coches azules”

The bilingual dictionary and the transfer rules of our toy rule-based system are presented in the Appendix.

The analysis and transfer actions are both executed by using the following command:
6.3 3-level transfer rules

So far, we have described how a toy rule-based system should be. Yet, in the Aper- tiun rule-based system the transfer rules are structured in a 3-level basis. The first level is called the 'chunk' level. Its goal is to detect and process patterns of words (chunks) and also translate word by word. For instance, in Spanish, the pattern formed by un\textless det\textgreater <ind\textless m\textgreater <sg\textgreater manzana\textless n\textgreater <f\textless pl\textgreater would be detected by a determiner–noun rule, which in this case would change the gender and number of the determiner so that it agrees with the noun; the result would be:

\begin{verbatim}
det_noun<SN<f<br> { un<det<br>ind<f<br>pl$ manzana<n<f<br>pl$ }$
\end{verbatim}

In other words, the first level gather the words of the sentence in syntagma, translates the words and processes rules within those syntagma. The second level, the inter-chunk processes rules between the syntagma. It reorders the chunks, modifies the “syntactic information” of each chunk and outputs them. For instance, if we have a sentence with the subject (SN) and the verb
(SV), the number and the person of the verb would be concorded with the number and person of the subject. Among other things, this is very helpful in the case of languages like Chinese where the verb goes sometimes at the end of the sentence. Trying to build one-level rules that matches the subject at the beginning of the sentence with the verb at the end of it, can be really difficult.

Finally, the last level is called the post-chunk. It sometimes carries out final tasks with the words inside the chunks and afterwards prints the whole sentence in a format accepted by the generator.
Chapter 7

Contributions and Further work

In this project, we have built an on-line translation Chinese-Spanish system together with two mobile applications. The translation paradigm behind them is the statistical MT. This paradigm is extremely convenient because it can be easily built when parallel corpora is available. However, the translation quality achieved may not be acceptable specially when translating out-of-domain texts. That is why, we have started to built a Chinese-to-Spanish rule-based MT system. This paradigm requires much more effort to be built and a lot of linguistic knowledge from the source and target language but it is more domain independent.

The most important contributions of this project are listed as follows:

- Chinese-Spanish web translator freely available at "http://www.chispa.me"
- Chinese-Spanish translator Android app freely available from Google Play
- Chinese-Spanish translator iOS app submitted to app store and waiting for approval.
- Research publications:
  Jordi Centelles, Marta R. Costa-jussà and Rafael E. Banchs Chispa on the Go, A mobile translation service for traveler in trouble. Submitted to the 14th European Chapter of the Association for Computational Linguistics Conference (EACL 2014). Gothenburg, Sweden, April, 2014
  Jordi Centelles, Marta R. Costa-jussà and Rafael E. Banchs. An IR-based strategy for supporting Chinese-Portuguese translation services in off-line mode. Submitted to Asia Information Retrieval Societies Conference, Singapore, December, 2013
- Open-source Chinese-Spanish rule-based MT system under development available at "http://www.apertium.org"
As future work, we aim to continue the project of the open-source Chinese-Spanish rule-based MT system in the context of the Google Summer of Code until the 15th of September 2013. By then, the rule-based MT system should be entirely built.

Additionally, we will continue maintaining and improving the app in order to keep increasing the number of downloads. One of the improvements that we are planning to include is an information retrieval system for supporting an offline mode within the app. It would be very useful when the user needs to translate something but he has no connection to the Internet.
Appendices
Appendix A

Research Publications
CHISPA on the GO
A mobile Chinese-Spanish translation service for travelers in trouble

Jordi Centelles\textsuperscript{1,2}, Marta R. Costa-jussà\textsuperscript{1,2} and Rafael E. Banchs\textsuperscript{2}
\textsuperscript{1} Universitat Politècnica de Catalunya, Barcelona
\textsuperscript{2} Institute for Infocomm Research, Singapore
\{visjcs,vismrc,rembanchs\}@i2r.a-star.edu.sg

Abstract

This demo showcases a translation service that allows travelers to have an easy and convenient access to Chinese-Spanish translations via a mobile app. The system integrates a phrase-based translation system with other open source components such as Optical Character Recognition and Automatic Speech Recognition to provide a very friendly user experience.

1 Introduction

During the last twenty years, Machine Translation technologies have matured enough to get out from the academic world and jump into the commercial area. Current commercially available machine translation services, although still not good enough to replace human translations, are able to provide useful and reliable support in certain applications such as cross-language information retrieval, cross-language web browsing and document exploration.

On the other hand, the increasing use of smartphones, their portability and the availability of internet almost everywhere, have allowed for lots of traditional on-line applications and services to be deployed on these mobile platforms.

In this demo paper we describe “CHISPA on the GO” a Chinese-Spanish translation service that intends to provide a portable and easy to use language assistance tool for travelers between Chinese and Spanish speaking countries.

The main three characteristics of the presented demo system are as follows:

• First, the system uses a direct translation between Chinese and Spanish, rather than using a pivot language as intermediate step as most of the current commercial systems do when dealing with distant languages.

• Second, in addition to support on-line translations, as other commercial systems, our system also supports access from mobile platforms, Android and iOS, by means of native mobile apps.

• Third, the mobile apps combine the base translation technology with other supporting technologies such as Automatic Speech Recognition (ASR), Optical Character Recognition (OCR), Image retrieval and Language detection in order to provide a friendly user experience.

2 SMT system description

The translation technology used in our system is based on the well-known phrase-based translation statistical approach (Koehn et al., 2003). This approach performs the translation splitting the source sentence in segments and assigning to each segment a bilingual phrase from a phrase-table. Bilingual phrases are translation units that contain source words and target words, and have different scores associated to them. These bilingual phrases are then selected in order to maximize a linear combination of feature functions. Such strategy is known as the log-linear model (Och and Ney, 2002). The two main feature functions are the translation model and the target language model. Additional models include lexical weights, phrase and word penalty and reordering.

2.1 Experimental details

Generally, Chinese-Spanish translation follows pivot approaches to be translated (Costa-jussà et al., 2012) because of the lack of parallel data to train the direct approach. The main advantage of our system is that we are using the direct approach and at the same time we rely on a pretty large corpus. For Chinese-Spanish, we use (1) the Holy Bible corpus (Banchs and Li, 2008), (2) the
United Nations corpus, which was released for research purposes (Rafalovitch and Dale, 2009), (3) a small subset of the European Parliament Plenary Speeches where the Chinese part was synthetically produced by translating from English, (4) a large TAUS corpus (TausData, 2013) which comes from technical translation memories, and (5) an in-house developed small corpus in the transportation and hospitality domains. In total we have 70 million words.

A careful preprocessing was developed for all languages. Chinese was segmented with Stanford segmenter (Tseng et al., 2005) and Spanish was preprocessed with Freeling (Padró et al., 2010). When Spanish is used as a source language, it is preprocessed by lower-casing and unaccented the input. Finally, we use the MOSES decoder (Koehn et al., 2007) with standard configuration: align-grow-final-and alignment symmetrization, 5-gram language model with interpolation and kneser-ney discount and phrase-smoothing and lexicalized reordering. We use our in-house developed corpus to optimize because our application is targeted to the travelers-in-need domain.

3 Web Translator and Mobile Application

This section describes the main system architecture and the main features of web translator and the mobile applications.

3.1 System architecture

Figure 1 shows a block diagram of the system architecture. Below, we explain the main components of the architecture, starting with the back-end and ending with the front-end.

3.1.1 Back-end

As previously mentioned, our translation system uses MOSES. More specifically, we use the open source MOSES server application developed by Saint-Amand (2013). Because translation tables need to be kept permanently in memory, we use binary tables to reduce the memory space consumption. The MOSES server communicates with a PHP script that is responsible for receiving the query to be translated and sending the translation back.

For the Chinese-Spanish language pair, we count with four types of PHP scripts. Two of them communicate with the web-site and the other two with the mobile applications. In both cases, one of the two PHP scripts supports Chinese to Spanish translations and the other one the Spanish to Chinese translations.

The functions of the PHP scripts responsible for supporting translations are: (1) receive the Chinese/Spanish queries from the front-end; (2) preprocess the Chinese/Spanish queries; (3) send these preprocessed queries to the Chinese/Spanish to Spanish/Chinese MOSES servers; (4) receive the translated queries; and (5) send them back to the front-end.

3.1.2 Front-end

HTML and Javascript constitute the main code components of the translation website. Another web development technique used was Ajax, which allows for asynchronous communication between the MOSES server and the website. This means that the website does not need to be refreshed after every translation.

The HTTP protocol is used for the communications between the web and the server. Specifically,
we use the POST method, in which the server receives data through the request message’s body.

The Javascript is used mainly to implement the input methods of the website, which are a Spanish keyboard and a Pinyin input method, both open source and embedded into our code. Also, using Javascript, a small delay was programmed in order to automatically send the query to the translator each time the user stops typing.

Another feature that is worth mentioning is the support of user feedback to suggest better translations. Using MySQL, we created a database in the server where all user suggestions are stored. Later, these suggestions can be processed off-line and used in order to improve the system.

Additionally, all translations processed by the system are stored in a file. This information is to be exploited in the near future, when a large number of translations has been collected, to mine for the most commonly requested translations. The most common translation set will be used to implement an index and search engine so that any query entered by a user, will be first checked against the index to avoid overloading the translation engine.

3.2 Android and iphone applications

The android app was programmed with the Android development tools (ADT). It is a plug-in for the Eclipse IDE that provides the necessary environment for building an app.

The Android-based “CHISPA on the GO” app is depicted in Figure 2.

For the communication between the Android app and the server we use the HttpClient interface. Among other things, it allows a client to send data to the server via, for instance, the POST method, as used on the website case.

For the Iphone app we use the xcode software provided by apple and the programming language used is Objective C.

In addition to the base translation system, the app also incorporates Automatic Speech Recognition (ASR), Optical Character Recognition technologies as input methods (OCR), Image retrieval and Language detection.

3.2.1 ASR and OCR

In the case of ASR, we relay on the native ASR engines of the used mobile platforms: Jelly-bean in the case of Android1 and Siri in the case of iOS2. Regarding the OCR implemented technology, this is an electronic conversion of scanned images into machine-encoded text. We adapted the open-source OCR Tesseract (released under the Apache license) (Tesseract, 2013).

3.2.2 Image retrieval

For image retrieving, we use the popular website flickr (Ludicorp, 2004). The image retrieving is activated with an specific button ”search Image” button in the app (see Figure 2). Then, an URL (using the HTTPClient method) is sent to a flickr server. In the URL we specify the tag (i.e. the topic of the images we want), the number of images, the secret key (needed to interact with flickr) and also the type of object we expect (in our case, a JSON object). When the server response is received, we parse the JSON object. Afterwards, with the HTTPConnection method and the information parsed, we send the URL back to the server and we retrieve the images requested. Also, the JAVA class that implements all these methods extends an AsyncTask in order to not block the user interface meanwhile is exchanging information with the flickr servers.

3.2.3 Language detection

We have also implemented a very simple but effective language detection system, which is very suitable for distinguishing between Chinese and Spanish. Given the type of encoding we are using

1http://www.android.com/about/jelly-bean/

2http://www.apple.com/ios/siri/
(UTF-8), codes for most characters used in Spanish are in the range from 40 to 255, and codes for most characters used in Chinese are in the range from 11,000 and 30,000. Accordingly, we have designed a simple procedure which computes the average code for the sequence of characters to be translated. This average value is compared with a threshold to determine whether the given sequence of characters represents a Chinese or a Spanish input.

4 Conclusions

In this demo paper, we described “CHISPA on the GO” a translation service that allows travelers-in-need to have an easy and convenient access to Chinese-Spanish translations via a mobile app.

The main characteristics of the presented system are: the use direct translation between Chinese and Spanish, the support of both website as well as mobile platforms, and the integration of supporting input technologies such as Automatic Speech Recognition, Optical Character Recognition, Image retrieval and Language detection.

As future work we intend to exploit collected data to implement an index and search engine for providing fast access to most commonly requested translations. The objective of this enhancement is twofold: supporting off-line mode and alleviating the translation server load.

Acknowledgments

The authors would like to thank the Universitat Politècnica de Catalunya and the Institute for Infocomm Research for their support and permission to publish this research. This work has been partially funded by the Seventh Framework Program of the European Commission through the International Outgoing Fellowship Marie Curie Action (IMTraP-2011-29951).

References


Appendix: Demo Script Outline

The presenter will showcase the “CHISPA on the GO” app by using the three different supported input methods: typing, speech and image. Translated results will be displayed along with related pictures of the translated items and/or locations when available. A poster will be displayed close to the demo site, which will illustrate the main architecture of the platform and will briefly explain the technology components of it.
An IR-based strategy for supporting Chinese-Portuguese translation services in off-line mode.

Jordi Centelles¹ ², Marta R. Costa-jussà¹ ², and Rafael E. Banchs ²

¹Universitat Politècnica de Catalunya, Barcelona
² Institute for Infocomm Research, Singapore
{visjc,vismrc,rembanchs}@i2r.a-star.edu.sg

Abstract. This paper describes an Information Retrieval engine that is used to support our Chinese-Portuguese machine translation services when no internet connection is available. Our mobile translation app, which is deployed on a portable device, relies by default on a server-based machine translation service, which is not accessible when no internet connection is available. For providing translation support under this condition, we have developed a contextualized off-line search engine that allows the users to continue using the app.

Keywords: online communications, structure, user generated content, emotions

1 Introduction

Machine translation applications have gained a lot of popularity in recent years. Currently, statistical approaches to machine translation are dominating the market, as they allow for automatically learning translation tables from parallel corpora (Brown et al 1993, Koehn et al 2003). The main problem for this approaches is the high amount of resources they consume regarding to memory and computational power. Due to this, most translation applications operate under a client-server architecture in which the client only provides a dummy interface while all the computations are carried out on a remote server. The main limitation of this scheme is that the client required internet connection to be available.

In this work, we present a search-based strategy for supporting machine translation services when internet connection is not available. More specifically, our proposed strategy, which is based on Information Retrieval technologies, is designed to support our Chinese-Portuguese translation service that has been deployed at the client side as a mobile app. The proposed strategy, allows for the mobile app to continued operating, with limited capabilities, on off-line mode when no internet connection is available. The off-line mode also includes contextualization strategies that allow improving the system performance based on user preferences, location and time.

The rest of the paper is structure as follows. In section 2, we describe the original Chinese-Portuguese on-line translation service. In section 3, we present the proposed
off-line mode strategy and its contextualization capabilities. Finally, in section 4, we present our conclusion and proposed future directions of research.

2 Chinese-Portuguese On-line Translation Services

In this section we describe the original Chinese-Portuguese on-line translation service. First, we present a brief overview on the Chinese-Portuguese machine translation engine (the server side), and then we present a detailed description of the mobile app that connects to this service (the client side).

2.1 Chinese-Portuguese Translation System

In order to build our machine translation system, we have used a standard phrase-based statistical machine translation based on Moses (Koehn et al., 2007). This well-known approach splits the source sentence to translate in segments and it assigns to each segment a bilingual phrase from a phrase-table. Bilingual phrases are translation units that contain source words and target words. These bilingual phrases have different scores associated to them (including conditional, posterior and lexical probabilities). Among the list of bilingual phrases, the decoder is in charge of selecting the ones that maximize the linear combination of feature functions. Such strategy is known as the log-linear model (Och and Ney, 2002). The two main feature functions are the translation model and the target language model. Additional models include phrase and word penalty and reordering.

Our system is a corpus-based approach where the key for translation quality is regarding the quality and quantity of the corpus used for training. Generally speaking, translation between distant language pairs follows pivot approaches through English (or other major-resourced language) because of the lack of parallel data to train the direct approach. The main advantage of our system is that we are using the direct approach and at the same time we rely on a pretty large corpus which has been properly preprocessed.

Regarding data preprocessing we have done the following:

- For Chinese, we have segmented the data using the Stanford Segmenter tool (Tseng et al., 2005).
- For Portuguese, we have true cased the data and tokenized it with Moses tools.

Moses was used with the standard configuration. Different training domain corpus where concatenated to a single training corpus. We have corpora from different domains available. In particular we have used the following ones:

- TAUS. Data provided by this organization include translation memories of technical content.
- In-house. This corresponds to a small corpus in the transportation and hospitality domains.
Corpus statistics for the training corpus are presented in Table 1. Just to give an idea of the quality of our translation system we report the automatic and human evaluation results for Chinese-Portuguese. For fine-tuning the translation engines, we have used the TAUS development dataset and, then, we have tested with the TAUS and In-house test. Results are shown in terms of the standard metric BLEU in Table 2.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Parameter</th>
<th>Chinese</th>
<th>Portuguese</th>
</tr>
</thead>
<tbody>
<tr>
<td>TAUS Train</td>
<td>Number of sentences</td>
<td>5 M</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Running words</td>
<td>57 M</td>
<td>62 M</td>
</tr>
<tr>
<td></td>
<td>Vocabulary</td>
<td>648 K</td>
<td>200 K</td>
</tr>
<tr>
<td>TAUS Dev</td>
<td>Number of sentences</td>
<td>808</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Running words</td>
<td>11 K</td>
<td>12 K</td>
</tr>
<tr>
<td></td>
<td>Vocabulary</td>
<td>3.0 K</td>
<td>3.4 K</td>
</tr>
<tr>
<td>TAUS Test</td>
<td>Number of sentences</td>
<td>721</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Running words</td>
<td>9.9 K</td>
<td>10.9 K</td>
</tr>
<tr>
<td></td>
<td>Vocabulary</td>
<td>2.8 K</td>
<td>3.3 K</td>
</tr>
<tr>
<td>In-house</td>
<td>Number of sentences</td>
<td>729</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Running words</td>
<td>4.1 K</td>
<td>4.7 K</td>
</tr>
<tr>
<td></td>
<td>Vocabulary</td>
<td>737</td>
<td>890</td>
</tr>
</tbody>
</table>

Table 1. Corpus details

<table>
<thead>
<tr>
<th>Translation direction</th>
<th>Domain / Dataset</th>
<th>Quality (BLEU)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chinese-to-Portuguese</td>
<td>TAUS</td>
<td>37.97</td>
</tr>
<tr>
<td></td>
<td>In-house</td>
<td>4.49</td>
</tr>
<tr>
<td>Portuguese-to-Chinese</td>
<td>TAUS</td>
<td>39.58</td>
</tr>
<tr>
<td></td>
<td>In-house</td>
<td>6.48</td>
</tr>
</tbody>
</table>

Table 2. Translation results

2.2 Chinese-Portuguese Translation App

The android app for the Chinese-Portuguese translation client was programmed with the Android development tools (ADT). It is a plug-in for the Eclipse IDE that provides the necessary environment for building an app.

The Android-based app is depicted in Figure 1. For the communication between the Android app and the server we use the HTTPClient interface. Among other things, it allows a client to send data to the server via, for instance, the POST method, as used on the website case.

In addition to the base translation system, the app also incorporates Automatic Speech Recognition (ASR), Optical Character Recognition technologies as input methods (OCR), Image retrieval and Language detection (Centelles et al., 2013).
Also, the system uses a database to store the translation performed by the system and keep track of the most used translations. To create the databases we used the popular open source database management system: MySQL.

3 Off-line Search-based Translation System

In this section we describe our proposed search-based off-line strategy to support the Chinese-Portuguese translation service. First, we describe our search engine implementation for translation, and then, we present the developed contextualization strategy for improving the performance of the system.

3.1 Search Engine for Translation

In most information retrieval applications the user provides a query aiming at recovering documents that are relevant to the query. The translation task can seen as conceptually similar, in the sense that the user provides a source sentence to be translated (a query) aiming at obtaining a meaningful translation for it.

In our proposed approach to translating by means of information retrieval we construct two composed indexes, one in each language, in which pointers to each other are also included. This index construction is performed in three steps:

- Common translation collection: we collect the most commonly Chinese and Portuguese sentences and their respective translations from the translation service. This
bilingual data collection is updated on a monthly basis according to the activity of the on-line registered users.

- Bilingual dictionary match: form the collected bilingual sentence pairs, a bilingual dictionary is used to identify Chinese and Portuguese term translations simultaneously occurring in the sentence pairs, which are replaced by entry codes in the dictionary. The entries of the used bilingual dictionary correspond with nouns and adjectives that are commonly observed in the translated pairs.

- A Chinese index is constructed by using the processed Chinese sentences and, in the same way, a Portuguese index is constructed by using the processed Portuguese sentences. The two indexes include pointers to each other so each Portuguese sentence points to its corresponding Chinese translation and each Chinese sentence points to its corresponding Portuguese translation.

These indexes are implemented by using the bag-of-words approach, for which the TF-IDF weighting scheme is used (Salton and Buckley 1988). For searching across the indexes, cosine similarity metric is used for ranking the retrieved outputs. Given a user input in the source language, the retrieval process is implemented in two steps:

- Dictionary match: the input sentence is evaluated for occurrences of terms from the bilingual dictionary. In case a term is detected, it is replaced by its corresponding entry code.
- Source search: two searches are performed over the source language index, the first one involves the original sentence provided by the user, and the second one involves the processed sentence (if terms have been found on it). The retrieved sentence with highest cosine similarity score is then selected.

Finally, the translation is constructed by using the corresponding sentence pair from the target language index:

- Sentence extraction: the target sentence corresponding to the selected source sentence is extracted from the target index if the obtained cosine similarity is high enough (current threshold value is 0.85).
- Sentence post edition: if the selected target sentence includes one or more dictionary entry codes on it, they are replaced by their corresponding dictionary forms before providing the final translation to the user.

Figure 2, illustrates the index construction, search, and translation generation processes used for the off-line translation system implementation.

### 3.2 Contextualized Translation Services

Finally, in this section we describe our contextualization strategy for improving the quality of the off-line translation service.

For providing the system with contextualization capabilities, each requested translation and its corresponding result from the online service are logged in the system along with the following types of metadata:
• User information: unique identification number for the user requesting the translation.

Fig. 2. Proposed approach for off-line translation by means of an information retrieval strategy over the collection of the most commonly requested translation pairs.

• Location information: spatial coordinates as provided by the GPS service of the mobile device at the moment the translation was requested.
• Time information: time stamp for the specific hour and day at which the translation was requested.
• Semantic information: a semantic categorization of the specific topic the requested translation belongs to.

These four types of metadata are used to train a personalized predictive model able to estimate which are the most probable translations the current user might be requesting in the next 24 hours, based on the current context (user-location-time) and previous translation history.

This model is updated every time the system is using the online mode, and the corresponding translation indexes and dictionaries are refreshed based on the model predictions. In this way, when going off-line, a personalized and contextualized translation service is locally available for the user.
4 Conclusions and Future Work

In this work we have described an Information Retrieval engine that is used to support our Chinese-Portuguese machine translation services when no internet connection is available. Our mobile translation app, which is deployed on a portable device, relies by default on a server-based machine translation service, which is not accessible when no internet connection is available. For providing translation support under this condition, we have developed a contextualized off-line search engine that allows the users to continue using the app.

As future work we plan to improve our off-line solution by incorporating predictive suggestions, so the system can suggest source sentences to the user by using partial inputs as queries for searching across the source index. We also want to improve the contextualization capabilities by including user dependent models for spatial and time localization.

Acknowledgements

The authors would like to thank their respective institutions: Institution 1 and Institution 2, for their support and permission for publishing this work.

References

Appendix B

Rule-based Toy System

B.1 Chinese monolingual dictionary

<?xml version="1.0" encoding="UTF-8"?>
<dictionary>
<sdefs>
  <sdef n="n"/>
  <sdef n="np"/>
  <sdef n="vblex"/>
  <sdef n="adj"/>
  <sdef n="part"/>
  <sdef n="count"/>
  <sdef n="sg"/>
  <sdef n="pl"/>
  <sdef n="prn"/>
</sdefs>

<pardefs>
  <pardef n="我...prn">
    <e>
      <p>
        <l/>
        <r><s n="prn"/></r>
      </p>
    </e>
  </pardef>

  [...]
</pardefs>
B.2 Spanish monolingual dictionary

```xml
<?xml version="1.0" encoding="UTF-8"?>
<dictionary>
  <sdefs>
    <sdef n="n" c="Noun"/>
    <sdef n="adj" c="Adjective"/>
    [...]  
    <sdef n="p1" c="First person"/>
    <sdef n="p2" c="Second person"/>
    <sdef n="p3" c="Third person"/>
  </sdefs>
  <pardefs>
    <pardef n="Jaime_np">  
    <e>
      <p>
        <l>
          <r><s n="np"/><s n="ant"/><s n="m"/><s n="sg"/></r>
          <p/>
          </l>
        </e>
      </pardef>
    <pardef n="/ser_vbser">  
    <e>
      <p>
        <l>somos</l>
        <r>ser<s n="vbser"/><s n="pri"/><s n="p1"/><s n="pl"/></r>
      </p>
    </pardef>
  </pardefs>
</dictionary>
```
<e>
<p>
<l>a</l>
<r>o<s n="det"/>s n="ind"/>s n="f"/>s n="sg"/></r>
</p>
</e>
<p>
<l>o</l>
<r>o<s n="det"/>s n="ind"/>s n="m"/>s n="pl"/></r>
</p>
</e>
<p>
<l>/</l>
<r>o<s n="det"/>s n="ind"/>s n="m"/>s n="sg"/></r>
</p>
</e>
</pardef>

<pardef n="niñ/o_u">
<e>
<p>
<l>as</l>
<r>o<s n="n"/>s n="f"/>s n="pl"/></r>
</p>
</e>
<p>
<l>a</l>
<r>o<s n="n"/>s n="f"/>s n="sg"/></r>
</p>
</e>
<p>
<l>o</l>
<r>o<s n="n"/>s n="m"/>s n="pl"/></r>
</p>
</e>
<p>
<l>o</l>
</p>
<r>o<s n="n"/><s n="m"/><s n="sg"/></r>
</p>
</e>
</pardef>

<pardef n="abstract/o_adj">
<p>
<l>as</l>
<r>o<s n="adj"/><s n="f"/><s n="pl"/></r>
</p>
</e>
</p>
</e>
</p>
<l>a</l>
<r>o<s n="adj"/><s n="f"/><s n="sg"/></r>
</p>
</e>
</p>
</e>
</p>
<l>os</l>
<r>o<s n="adj"/><s n="m"/><s n="pl"/></r>
</p>
</e>
</p>
</e>
</p>
<l>o</l>
<r>o<s n="adj"/><s n="m"/><s n="sg"/></r>
</p>
</e>
</p>
</e>
</p>
<l>ísimas</l>
<r>o<s n="adj"/><s n="sup"/><s n="f"/><s n="pl"/></r>
</p>
</e>
</p>
</e>
</p>
</l>ísimas</l>
<r>o<s n="adj"/><s n="sup"/><s n="f"/><s n="sg"/></r>
</p>
</e>
</p>
</e>
</p>
</l>ísimas</l>
<r>o<s n="adj"/><s n="sup"/><s n="f"/><s n="sg"/></r>
</p>
</e>
</p>
</e>
</p>
</l>ísimas</l>
B.3 Bilingual dictionary

<?xml version="1.0" encoding="UTF-8"?>
<dictionary>
<sdefs>
<sdef n="n" c="Noun"/>
<sdef n="np" c="Proper noun"/>
<sdef n="adj" c="Adjective"/>
<sdef n="f" c="Feminine"/>
<sdef n="m" c="Masculine"/>
<sdef n="mf" c="Masculine / feminine"/>
<sdef n="sg" c="Singular"/>
<sdef n="pl" c="Plural"/>
<sdef n="vbser" c="Verb 'to be'"/>
<sdef n="ant" c="Anthroponym"/>
</dictionary>
<sdef n="ind" c="Indefinite"/>
<sdef n="vblex" c="Verb"/>
<sdef n="det" c="Determiner"/>
</sdefs>

<section id="main" type="standard">
<p>
小明<s n="np"/>
Jaime<s n="np"/>
是<s n="vblex"/>

一<s n="det"/>
uno<s n="det"/>
小<s n="adj"/>

男孩<s n="n"/>

男<s n="m"/>
<s n="sg"/>
<p>
</p>
</section>

B.4 Transfer rules

<?xml version="1.0" encoding="UTF-8"?>
<transfer>

<section-def-cats>
<def-cat n="nom"/>
<cat-item tags="n"/>
</def-cat>

<def-cat n="nom_personal"/>
<cat-item tags="np"/>
</def-cat>

<def-cat n="adjectius"/>
<cat-item tags="adj"/>
</def-cat>

<def-cat n="determinant"/>
<cat-item tags="det"/>
</def-cat>

<def-cat n="vrb"/>
<cat-item tags="vblex"/>
<cat-item tags="vbser"/>
</def-cat>
</transfer>
</section-def-cats>

<section-def-attrs>
<def-attr n="a_nom">
<attr-item tags="n"/>
<attr-item tags="n.acr"/>
<attr-item tags="np.loc"/>
<attr-item tags="np.ant"/>
<attr-item tags="np.cog"/>
<attr-item tags="np.al"/>
<attr-item tags="adv"/>
[...]
</def-attr>
</section-def-attrs>

<section-rules>

<rule>
<pattern>
<pattern-item n="nom_personal"/>
<pattern-item n="vrb"/>
</pattern>
</action>
</rule>
}</section-rules>
<rule>
  <pattern>
    <pattern-item n="determinant"/>
    <pattern-item n="adjectius"/>
    <pattern-item n="nom"/>
  </pattern>
  <action>
    <out>
      <lu>
        <clip pos="1" side="tl" part="lem"/>
        <clip pos="1" side="tl" part="a_det"/>
        <clip pos="3" side="tl" part="gen"/>
        <clip pos="3" side="tl" part="nbr"/>
      </lu>
      <b/>
      <lu>
        <clip pos="3" side="tl" part="lem"/>
        <clip pos="3" side="tl" part="a_nom"/>
        <clip pos="3" side="tl" part="gen"/>
        <clip pos="3" side="tl" part="nbr"/>
      </lu>
      <b/>
      <lu>
        <clip pos="2" side="tl" part="lem"/>
        <clip pos="2" side="tl" part="a_adj"/>
        <clip pos="3" side="tl" part="gen"/>
        <clip pos="3" side="tl" part="nbr"/>
      </lu>
    </out>
  </action>
</rule>
Bibliography


   http://www.statmt.org/moses/?n=moses.webtranslation.


