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

Jordi Centelles[1,2], **Marta R. Costa-jussà**[1,2] **and Rafael E. Banchs**[2]
[1] Universitat Politècnica de Catalunya, Barcelona
[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

Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics, pages 33–36,
Gothenburg, Sweden, April 26-30 2014. ©2014 Association for Computational Linguistics

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.

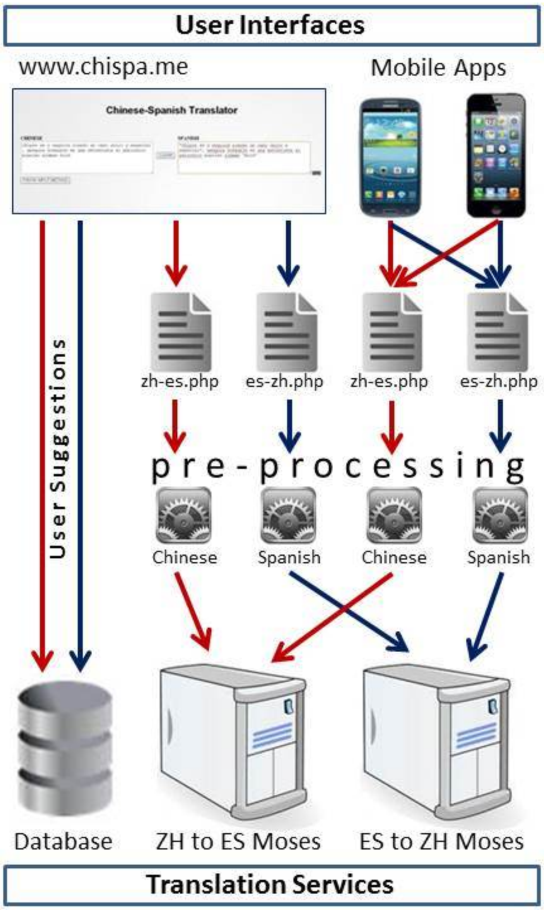


Figure 1: Block diagram of the system architecture

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.

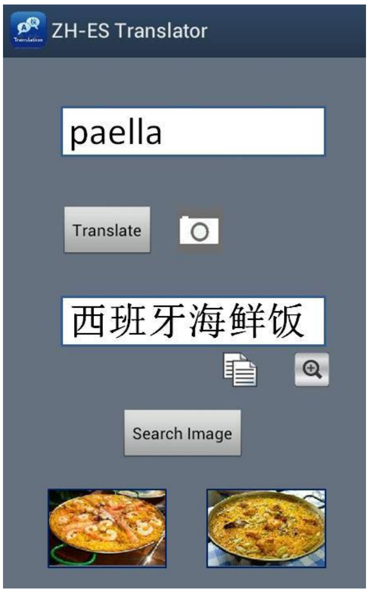
Another feature that is worth mentioning is the support of user feedback to suggest better translations. Using MYSQL, we created a database in the server where all user suggestions are stored. Later, these suggestions can be processed off-line and used in order to improve the system.

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.


ZH-ES Translator

paella

西班牙海鲜饭

Figure 2: Android application

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 Android[1] and Siri in the case of iOS[2]. 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

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

[2] http://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) and the HLT Department of the Institute for Infocomm Reseach.

References

R. E. Banchs and H. Li. 2008. Exploring Spanish Morphology effects on Chinese-Spanish SMT. In *MATMT 2008: Mixing Approaches to Machine Translation*, pages 49–53, Donostia-San Sebastian, Spain, February.

M. R. Costa-jussà, C. A. Henríquez Q, and R. E. Banchs. 2012. Evaluating indirect strategies for chinese-spanish statistical machine translation. *J. Artif. Int. Res.*, 45(1):761–780, September.

P. Koehn, F.J. Och, and D. Marcu. 2003. Statistical Phrase-Based Translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL'03).*

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL'07)*, pages 177–180, Prague, Czech Republic, June.

Ludicorp. 2004. Flickr. accessed online May 2013 `http://www.flickr.com/`.

F.J. Och and H. Ney. 2002. Dicriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02)*, pages 295–302, Philadelphia, PA, July.

L. Padró, M. Collado, S. Reese, M. Lloberes, and I. Castellón. 2010. FreeLing 2.1: Five Years of Open-Source Language Processing Tools. In *Proceedings of 7th Language Resources and Evaluation Conference (LREC 2010)*, La Valleta, Malta, May.

A. Rafalovitch and R. Dale. 2009. United Nations General Assembly Resolutions: A Six-Language Parallel Corpus. In *Proceedings of the MT Summit XII*, pages 292–299, Ottawa.

H. Saint-Amand. 2013. Moses server. accessed online May 2013 `http://www.statmt.org/moses/?n=Moses.WebTranslation`.

TausData. 2013. Taus data. accessed online May 2013 `http://www.tausdata.org`.

Tesseract. 2013. Ocr. accessed online May 2013 `https://code.google.com/p/tesseract-ocr/`.

H. Tseng, P. Chang, G. Andrew, D. Jurafsky, and C. Manning. 2005. A conditional random field word segmenter. In *Fourth SIGHAN Workshop on Chinese Language Processing*.

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.