Morphology Generation for Statistical Machine Translation using Deep Learning Techniques

Anonymous ACL submission

Abstract

Morphology in unbalanced languages remains a big challenge in the context of machine translation. In this paper, we propose to de-couple machine translation from morphology generation in order to better deal with the problem. We investigate the morphology simplification with a reasonable trade-off between expected gain and generation complexity. For the Chinese-Spanish task, optimum morphological simplification is in gender and number. For this purpose, we design a new classification architecture which, compared to other standard machine learning techniques, obtains the best results. This proposed neural-based architecture consists of several layers: an embedding, a convolutional followed by a recurrent neural network and, finally, ends with sigmoid and softmax layers. We obtain classification results over 98% accuracy in gender classification, over 93% in number classification, and an overall translation improvement of 0.7 METEOR.

1 Introduction

Machine Translation (MT) is evolving from different perspectives. One of the most popular paradigms is still Statistical Machine Translation (SMT), which consists in finding the most probable target sentence given the source sentence using probabilistic models based on co-ocurrences. Recently, deep learning techniques applied to natural language processing, speech recognition and image processing and even in MT have reached quite successful results. Early stages of deep learning applied to MT include using neural language modeling for rescoring (Schwenk et al., 2007). Later, deep learning has been integrated in MT in many different steps (Zhand and Zong, 2015). Nowadays, deep learning has allowed to develop an entire new paradigm, which within one-year of development has achieved state-of-the-art results (Jean et al., 2015) for some language pairs.

In this paper, we are focusing on a challenging translation task, which is Chinese-to-Spanish. This translation task has the characteristic that we are going from an isolated language in terms of morphology (Chinese) to a fusional language (Spanish). This means that for a simple word in Chinese (e.g. 鼓励), the corresponding translation has many different morphology inflexions (e.g. *alentar, alienta, alentamos, alientan* ...), which depend on the context. It is still difficult for MT in general (no matter which paradigm) to extract information from the source context to give the correct translation.

We propose to divide the problem of translation into translation and then a postprocessing of morphology generation. This has been done before, e.g. (Toutanova et al., 2008; Formiga et al., 2013), as we will review in the next section. However, the main contribution of our work is that we are using deep learning techniques in morphology generation. This gives us significant improvements in translation quality.

The rest of the paper is organised as follows. Section 2 describes the related work both in morphology generation approaches and in Chinese-Spanish translation. Section 3 overviews the phrase-based MT approach together with an explanation of the divide and conquer approach of translating and generating morphology. Section 4 details the architecture of the morphology generation module and it reports the main classification techniques that are used for morphology generation. Section 5 describes the experimental framework. Section 6 reports and discusses both classification and translation results, which show significant improvements. Finally, section 7 summarises the main conclusions and further work.

2 Related Work

In this section we are reviewing the previous related works on morphology generation for MT and on Chinese-Spanish MT approaches.

Morphology generation There have been many works in morphological generation and some of them are in the context of the application of MT. In this cases, MT is faced in two-steps: first step where the source is translated to a simplified target text that has less morphology variation than the original target; and then, second step, a postprocessing module (morphology generator) adds the proper inflections. To name a few of these works, for example, (Toutanova et al., 2008) build maximum entropy markov models for inflection prediction of stems; (Clifton and Sarkar, 2011) and (Kholy and Habash, 2012) use conditional random fields (CFR) to predict one or more morphological features; and (Formiga et al., 2013) use Support Vector Machines (SVMs) to predict verb inflections. Other related works are in the context of Part-of-Speech (PoS) tagging generation such as (Giménez and Màrquez, 2004) in which a model is trained to predict each individual fragment of a PoS tag by means of machine learning algorithms. The main difference is that in PoS tagging the word itself has information about morphological inflection, whereas in our task, we do not have this information.

In this paper, we use deep learning techniques to morphology generation or classification. Based on the fact that Chinese does not have number and gender inflections and Spanish does, (Costa-jussà, 2015) show that simplification in gender and number has the best trade-off between improving translation and keeping the morphology generation complexity at a low level.

Chinese-Spanish There are few works in Chinese-Spanish MT despite being two of the most spoken languages in the world. Most of these works are based on comparing different pivot strategies like standard cascade or pseudo-corpus (Costa-jussà et al., 2012). Also it is important to mention that, in 2008, there were two tasks organised by the popular IWSLT evaluation campaign[1] (International Workshop on Spoken Language Translation) between these two languages (Paul, 2008). The first task was based on a direct translation for Chinese-Spanish. The second task provided corpus in Chinese-English and English-Spanish and asked participants to provide Chinese-Spanish translation through pivot techniques. The second task obtained better results than direct translation because of the larger corpus provided. Differently, (Costa-jussà and Centelles, 2016) present the first rule-based MT system for Chinese to Spanish. Authors describe a hybrid method for constructing this system taking advantage of available resources such as parallel corpora that are used to extract dictionaries and lexical and structural transfer rules. Finally, it is worth mentioning that novel successful neural approximations (Jean et al., 2015), already mentioned in the introduction, have not yet achieved state-of-the-art results for this language pair (Costa-jussà et al., 2017).

3 Machine Translation Architecture

In this section, we review the baseline system which is a standard phrase-based MT system and explain the architecture that we are using by dividing the problem of translation into: morphologically simplified translation and morphology generation.

3.1 Phrase-based MT baseline system

The popular phrase-based MT system (Koehn et al., 2003) focuses on finding the most probable target text given a source text. In the last 20 years, the phrase-based system has dramatically evolved introducing new techniques and modifying the architecture; for example, replacing the noisy-channel for the log-linear model which combines a set of feature functions in the decoder, including the translation and language model, the reordering model and the lexical models. There is a widely used open-source software, *Moses* (Koehn et al., 2007), which englobes a large community that helps in the progress of the system. As a consequence, phrase-based MT is a commoditized technology used at the academic and commercial level. However, there are still many challenges to solve, such as morphology generation.

[1] http://iwslt2010.fbk.eu

3.2 Divide and conquer MT architecture: simplified translation and morphology generation

Morphology generation is not always achieved in the standard phrase-based system. This may be due to the fact that phrase-based MT uses a limited source context information to translate. Therefore, we are proposing to follow a similar strategy to previous works (Toutanova et al., 2008; Formiga et al., 2013), where authors do a first translation from source to a morphology-based simplified target and then, use the morphology generation module that transforms the simplified translation into the full form output.

4 Morphological Generation Module

In order to design the morphology generation module, we have to decide the morphology simplification we are applying to the translation. Since we are focusing on Chinese-to-Spanish task and based on (Costa-jussà, 2015), the simplification which achieves the best trade-off among highest translation gain and lowest complexity of morphological generation is the simplification in number and gender. Table 1 shows examples of this simplification. The main challenge of this task is that number and gender are generated from words where this inflection information (both number and gender) has been removed beforehand.

With these results at hand, we propose an architecture of the morphology generation module, which is language independent and it is easily generalizable to other simplification schemes.

The morphology generation procedure is summarised as follows and further detailed in the next subsections.

- Feature selection. We have investigated different set of features including information from both source and target languages.

- Classification. We propose a new deep learning classification architecture composed of different layers.

- Rescoring and rules. We generate different alternatives of the classification output and rerank them using a language model. After, we use hand-crafted rules that allow to solve some specific problems.

This procedure is depicted in Figure 1, in which we can see that each of the above processes generates the needed input for the next step. Figure also shows in red the main subprocesses that have been developed on this work.

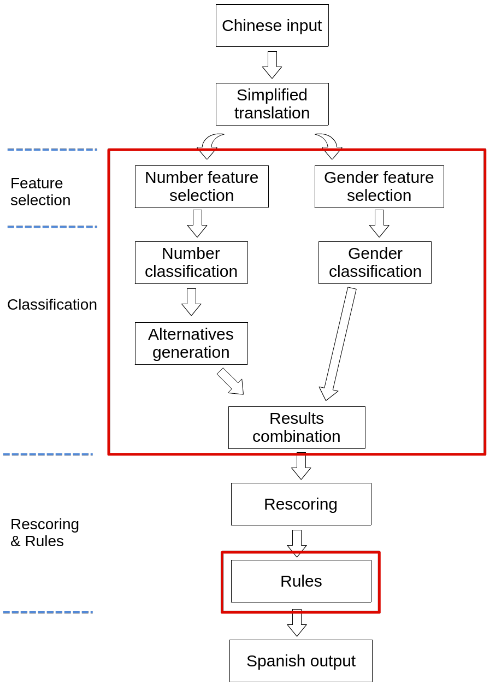

Figure 1: Block Diagram for Morphology Generation

4.1 Feature selection

We propose to compare several features for recovering morphological information. Given that both Chinese and simplified Spanish languages do not contain explicit morphology information, we start by simply using windows of words as source of information. We follow Collobert's approach in which each word is represented by a fixed size window of words in which the central element is the one to classify (Collobert et al., 2011).

In our case, we experiment with three different inputs: (1) Chinese window; (2) simplified Spanish window; (3) Spanish window adding information about its correspondant word in Chinese, i.e. information about pronouns and the number of characters in the word. The main advantage of the second one is that it is not dependant on the alignment file generated during translation.

Our classifiers did not have to train all types of words. Some types of words, such as prepositions (*a, ante, cabo, de...*), do not have gender or number. Therefore our system was trained using only determiners, adjectives, verbs, pronouns and nouns which are the ones that present morphology

Es$_{num}$	decidir[VMIP3**N**0] examinar[VMN0000] el[DA0M**N**0] c uestión[NCF**N**000] en[SPS00] el[DA0M**N**0] período[NCM **N**000] de[SPS00] sesión[NCF**N**000] el[DA0M**N**0] tema[NCM**N**000] titular [AQ0M**N**0] “[Fp] cuestión[NCF**N**000] relativo[AQ0F**N**0] a[SPS00] el[DA0M**N**0] derecho[NCM**N**000] humano[AQ0M**N**0] “[Fp] .[Fp]
Es$_{gen}$	decidir[VMIP3S0] examinar[VMN0000] el[DA0**G**S0] cuestión [NC**G**S000] en[SPS00] el[DA0**G**S0] período[NC**G**S000] de [SPS00] sesión[NC**G**S000] el[DA0**G**S0] tema[NC**G**S000] titular [AQ0**G**S0] “[Fp] cuestión[NC**G**S000] relativo[AQ0**G**S0] a[SPS00] el[DA0**G**S0] derecho[NC**G**S000] humano[AQ0**G**S0] “[Fp] .[Fp]
Es$_{numgen}$	decidir[VMIP3**N**0] examinar[VMN0000] el[DA0**GN**0] cuestión[NC**GN**000] en[SPS00] el[DA0**GN**0] período[NC**GN**000] de[SPS00] sesión[NC**GN**000] el[DA0**GN**0] tema[NC**GN**000] titular [AQ0**GN**0] “[Fp] cuestión[NC**GN**000] relativo[AQ0**GN**00] a[SPS00] el[DA0**GN**0] derecho[NC**GN**000] humano[AQ0**GN**0] “[Fp] .[Fp]
Es	Decide examinar la cuestión en el período de sesiones el tema titulad o “ Cuestiones relativas a los derechos humanos ” .

Table 1: Example of Spanish simplification into number, gender and both

.

variations in gender or number. However, note that all types of words are used in the windows.

4.2 Classification architecture

Description We propose to train two different models: one to retrieve gender and another to retrieve number. Each model decides among three different classes. Classes for gender classifier are masculine (M), femenine (F) and none (N); and classes for number classifier are singular (S), plural (P) and none (N) [2]. Again, we inspire our architecture in previous Collobert's proposal and we modify it by adding a recurrent neural network. This recurrent neural network is relevant because it keeps information about previous elements in a sequence and, in our classification problem, context words are very relevant. As a recurrent neural network, we use a Long Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) that is proven efficient to deal with sequence NLP challenges (Sutskever et al., 2014). This kind of recurrent neural network is able to maintain information for several elements in the sequence and to forget it when needed. Figure 2 shows an overview of the different layers involved in the final classification architecture, which are detailed as follows:

Embedding. We represent each word as its index in the vocabulary, i.e. every word is represented as one discrete value:

$E(w) = W, W \in \mathbb{R}^d$,

w being the index of the word in the sorted vocabulary and d the user chosen size of the array. Then, each word is represented as a numeric array and each window is a matrix.

Convolutional. We add a convolutional neural network. This step allows the system to detect some common patterns between the different words. This layer's input consists in W^l matrix of multidimensional arrays of size $n \cdot d$, where n is the window length (in words) and d is the size of the array created by the previous embedding layer. This layer's output is a matrix of the same size as the input.

Max Pooling. This layer allows to extract most relevant features from the input data and reduces feature vectors to half.

LSTM. Each feature array is treated individually, generating a fixed size representation h_i of the ith word using information of all the previous words (in the sequence). This layer's output, h, is the result of the last element of the sequence using information from all previous words.

Sigmoid. This layer smoothes results obtained by previous layer and compresses results to the interval $[-1, 1]$. This layer's input is a fixed size vector of shape $1 \cdot n$ where n is the number of neurons in the previous LSTM layer. This layer's output is a vector of length c equal to the number of classes to predict.

Softmax. This layer allows to show results as probabilities by ensuring that the returned value of each class belongs to the $[0, 1)$ interval and all classes add up 1.

Motivation Our input data is PoS tagged and morphogically simplified before the classification architecture which largely reduces the information that can be extracted from individual words in the vocabulary. In addition, we can encounter out-of-vocabulary words for which no morphological information can be extracted.

The main source of information available is the context in which the word can be found in the sentence. Considering the window as a sequence enforces the behaviour a human would have while reading the sentence. The information of a word consists in itself and the words that surround it. Sometimes information preceeds the word and sometimes information is after the word. Words

[2] *None* means that there is no need to specify gender or number information because the word is invariant in these terms. This happens for types of words (determiners, nouns, verbs, pronouns and adjectives) that can have gender and number in other cases.

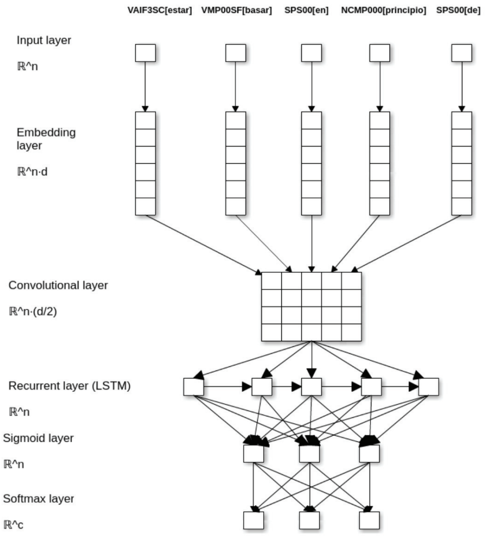

Figure 2: Neural network overview.

(like adjetives), which are modifying or complementing another word, generally take information from preceeding words. For example, in the sequence *casa blanca*, the word *blanca* could also be *blanco, blancos or blancas* but because noun and adjective are required to have gender and number agreement, the femenine word *casa* forces the femenine for *blanca*. While, for example, determiners usually take information from posterior words. This fact motivates that the word to classify has to be placed at the center of the window.

Finally, given that we rely only on the context information since words themselves may not have any information, makes the recurrent neural network a key element in our architecture. The output h of the layer can be considered a context vector of the whole window maintaining information of all the previously encountered words (in the same window).

4.3 Rescoring and rules

At this point in the pipeline, we have two models (gender and number) that allow us to generate the full Part-of-Speech (PoS) tag by combining the best results of both classifiers.

However, in order to improve the overall performance, we add a rescoring step followed by some hand-crafted rules. To generate the different alternatives, we represent every sentence as a graph (see Figure 3), with the following properties:

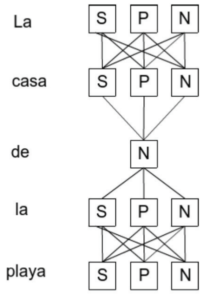

Figure 3: Example of sentence graph. S stands for singular, P for plural and N for none

- Each word is represented as a layer of the graph and each node represents a class of the classification model.

- A node only has edges with all the nodes of the next layer. This way we force the original sentence order.

- An edge's weight is the probability given by the classification model.

- Each accepted path in the graph starts in the first layer and ends in the last one. This acyclic structure finds the best path in linear time, due to the fact that it goes through all layers and it picks the node with the greatest weight. One layer can have either 1 element (the word does not need to be classified, e.g. prepositions) or 3 elements (the word needs to be classified among the three number or gender categories).

- Add the weight of a previously trained target language model.

We used Yen's algorithm (Yen, 1971) to find the best path, which has an associated cost of $O(KN^2 logN)$, being K the number of paths to find.

See pseudo code in Algorithm 1, where A is the set paths chosen in the graph. The algorithm ends when this A set contains K paths or no further paths available to explore. B contains all suboptimal paths that can be elected in future iterations.

There are two special cases that the models were not able to treat and we apply specific rules: (1) conjunctions *y* and *o* are replaced by *e* and *u* if they are placed in front of vowels. This could not be generated during translation because both words share the same tag and lemma; (2) verbs with a pronoun as a suffix, *producirse*, second to last syllabe stretched (*palabras llanas*) and ending in a vowel are not accentuated. However, after adding the suffix, these words should be accentuated because they become *palabras esdrújulas*, which happen to be always accentuated.

```
Data: G Graph of the sentence, K
Result: best k paths in G
initialization;
A[0] = bestPath(G,0,final);
B = [] ;
i = 0;
for i <K do
    for  i in range(0, len(A[K-1])-1) do
        spurNode = A[K-1][i];
        root = A[K-1][0;i];
        for path in A do
            if root = path[0:i] then
                remove edge(i-1,i) from G;
            end
        end
        for node in root and node != spurNode do
            removes node node from G;
        end
        spurPath = bestPath(G,spurnode, final) totalPath = root +
        spurPath;
        B.append(totalPath);
        restore edges from G;
        restore nodes from G;
        if  B is empty then
            break;
        end
        B.sort();
        A.append(B[0]);
        B.pop();
    end
end
```

Algorithm 1: Pseudo-code for k-best paths generation.

5 Experimental framework

In this section, we describe the data used for experimentation together with the corresponding preprocessing. In addition, we detail chosen parameters for the MT system and the classification algorithm.

5.1 Data and preprocessing

One of the main contributions of this work is using the Chinese-Spanish language pair. In the last years, there has appeared more and more resources for this language pair available in (Ziemski et al., 2016) or from TAUS corporation[3]. Therefore, differently from previous works on this language pair, we can test our approach in both a small and large data sets.

L	Set		S	W	V
ES	Train	Small	58.6K	2.3M	22.5K
		Large	3.0M	51.7M	207.5K
	Development		990	43.4K	5.4k
	Test		1K	44.2K	5.5K
ZH	Train	Small	58.6K	1.6M	17.8K
		Large	3.0M	43.9M	373.5K
	Development		990	33K	3.7K
	Test		1K	33.7K	3.8K

Table 2: Corpus Statistics. Number of sentences (S),words (W), vocabulary (V). M stands for millions and K stands for thousands.

- A small training corpus by using the United Nations Corpus (UN) (Rafalovitch and Dale, 2009).

- A large training corpus by using, in addition to the UN corpus, the TAUS corpus, the Bible corpus (Chew et al., 2006) and the BTEC (Basic Traveller Expressions Corpus) (Takezawa, 2006). The TAUS corpus is around 2,890,000 sentences, the Bible corpus about 30,000 sentences and the BTEC corpus about 20,000 sentences.

Corpus statistics are shown in Table 2. Development and test sets are taken from UN corpus.

Corpus preprocessing consisted in tokenization, filtering empty sentences and longer than 50 words, Chinese segmentation by means of the ZhSeg (Dyer, 2016), Spanish lowercasing, filtering pairs of sentences with more than 10% of non-Chinese characters in the Chinese side and more than 10% of non-Spanish characters in the Spanish side. Spanish PoS tagging was done using *Freeling* (Padró and Stanilovsky, 2012). All chunking or name entity recognition was disabled to preserve the original number of words.

5.2 MT Baseline

Moses has been trained using default parameters, which include: grow-diag-final word alignment symmetrization, lexicalized reordering, relative frequencies (conditional and posterior probabilities) with phrase discounting, lexical weights, phrase bonus, accepting phrases up to length 10, 5-gram language model with kneser-ney smoothing, word bonus and MERT optimisation.

5.3 Classification parameters

To generate the classification architecture we used the library *keras* (Chollet, 2015) for creating and ensambling the different layers. Using NVIDIA GTX Titan X GPUs with 12GB of memory and 3072 CUDA Cores, each classifier is trained on aproximately 1h and 12h for the small and large corpus, respectively.

[3] http://www.taus.net

Regarding classification parameters, experimentation has shown that number and gender classification tasks have different requirements. Table 3 summarizes these parameters. The best window size is 9 and 7 words for number and gender, respectively. In both cases increasing this size lowers the accuracy of the system. The vocabulary size is fixed as a trade-off between giving enough information to the system to perform the classification while removing enough words to train the classifier for unknown words. The embedding size of 128 results in stable training, while further increasing this value augmented the training time and hardware cost. The filter size in the convolutional layer reached best results when it was slightly smaller than the window size, being 7 and 5 the best values for number and gender classification, respectively. Finally, increasing LSTM nodes up to 70 improved significantly for both classifiers.

Table 3: Values of the different parameters of the classifiers

Parameter	Small		Large	
	Num	Gen	Num	Gen
Window size	9	7	9	7
Vocabulary size	7000	9000	15000	15000
Embedding	128	128	128	128
Filter size	7	5	7	5
LSTM nodes	70	70	70	70

For windows, we only used the simplified Spanish translation. In Table 4 we can see that testing different sources of information with the classifier of number for the small corpus. Adding Chinese has a negative effect in the classifier accuracy.

5.4 Rescoring and Full form generation

As a rescoring tool, we use the one available in *Moses* [4]. We trained a standard n-gram language model with the SRILM toolkit (Stolcke, 2002).

In order to generate the final full form we use the full PoS tag, generated from the post-processing step, and the lemma, taken from the morphology-simplified translation output. Then, we use the vocabulary and conjugations rules provided by *Freeling*. *Freeling*'s coverage raises the 99%. When a word is not found in the dictionary, we test all gender and/or number inflections in descendant order of probability until a match is found. If none matched, the lemma is used as translation, which usually happens only in the case of cities or demonyms.

Table 4: Accuracy of the classifier of number using different sources of information.

Features	Accuracy (%)
Chinese window	72
Spanish window	93,4
Chinese + Spanish window	86

6 Evaluation

In this section we discuss the results obtained both in classification and in the final translation task.

Table 5 shows results for the classification task both number and gender and with the different corpus sets. We have contrasted our proposed classification architecture based on neural networks with standard machine learning techniques such as linear, cuadratic and sigmoid kernels SVMs (Cortes and Vapnik, 1995), random forests (Breiman, 2001), convolutional(Fukushima, 1980) and LSTM(Hochreiter and Schmidhuber, 1997) neural networks (NN). All algorithms were tested using features and parameters described in previous sections with the exception of random forests in which we added the one hot encoding representation of the words to the features.

We observe that our proposed architecture achieves by large the best results in all tasks. It is also remarkable that the accuracy is lower using the bigger corpus, this is due to the fact that the small set consisted in texts of the same domain and the vocabulary had a better representation of specific words such as country names.

Table 5: Classification results. In bold, best results. Num stands for Number and Gen, for Gender

Algorithm	Small		Large	
	Num	Gen	Num	Gen
Naive Bayes	61.3	53.5	61.3	53.5
Lineal kernel SVM	68.1	71.7	65.8	69.3
Cuadratic kernel SVM	77.8	81.3	77.6	82.7
Sigmoid kernel SVM	83,1	87.4	81.5	84.2
Random Forest	81.6	91.8	77.8	88.1
Convolutional NN	81.3	93.9	83.9	94.2
LSTM NN	68.1	73.3	70.8	71.4
CNN + LSTM	**93.7**	**98.4**	**88.9**	**96.1**

Tabla 6 shows translation results. We show both the Oracle and the result in terms of METEOR (Banerjee and Lavie, 2005). We observe improvement in most cases (when classifying number, gender, both and rescoring), but best results are obtained when classifying number and gender and rescoring number in the large corpus, obtaining a

[4] https://github.com/moses-smt/mosesdecoder/tree/master/scripts/nbest-rescore

Table 6: METEOR results. In bold, best results. Num stands for Number and Gen, for Gender

Set	System	UN	
		Oracle	Result
Small	Baseline	-	55.29
	+Num	55.60	55.35
	+Gen	55.45	55.39
	+Num&Gen	56.81	55.48
	+Num&Gen +Rescoring(Num&Gen)	-	54.91
	+Num&Gen +Rescoring(Num)	-	**55.56**
Large	Baseline	-	56.98
	+Num	58.87	57.51
	+Gen	57.56	57.32
	+Num&Gen	62.41	57.13
	+Num&Gen Rescoring	-	**57.74**

gain up to +0.7 METEOR.

Rescoring step improves final results. Note that rescoring was only applied to number classification because gender classification model has a low classification error (bellow 2%) which makes it harder to further decrease it. Additionally, gender and number classification scores are not be comparable and not easily integrated in Yen's algorithm.

7 Conclusions

Chinese-to-Spanish translation task is challenging, specially because of Spanish being morphologically rich compared to Chinese. Main contributions of this paper include correctly de-coupling the translation and morphological generation tasks and proposing a new classification architecture, based on deep learning, for number and gender.

Standard phrase-based MT procedure is changed to first translating into a morphologically simplified target (in terms of number and gender); then, introducing the classification algorithm, based on a new proposed neural network-based architecture, that retrieves the simplified morphology; and composing the final full form by using the standard *Freeling* dictionary.

Results of the proposed neural-network architecture in the classification task compared to standard algorithms (SVM or random forests) are significantly better and results in the translation task achieve up to 0.7 METEOR improvement. As further work, we intend to further simplify morphology and extend the scope of the classification.

Acknowledgements

References

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, volume 29, pages 65–72.

Leo Breiman. 2001. Random forests. *Machine learning*, 45(1):5–32.

Peter A Chew, Steve J Verzi, Travis L Bauer, and Jonathan T McClain. 2006. Evaluation Of The Bible As A Resource For Cross-language Informatio n Retrieval. In *Proceedings of the Workshop on Multilingual Language Resources a nd Interoperability*, pages 68–74.

François Chollet. 2015. Keras, https://github.com/fchollet/keras.

Ann Clifton and Anoop Sarkar. 2011. Combining morpheme-based machine translation with post-processing morpheme prediction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 32–42, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.

Marta R. Costa-jussà and Jordi Centelles. 2016. Description of the chinese-to-spanish rule-based machine translation system developed using a hybrid combination of human annotation and statistical techniques. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 15.

Marta R. Costa-jussà, Carlos A. Henríquez Q., and Rafael E. Banchs. 2012. Evaluating indirect strategies for chinese-spanish statistical machine translation. *Journal Of Artificial Intelligence Research*, 45:762–780.

Marta R. Costa-jussà, David Aldón, and José A. R. Fonollosa. 2017. Chinese-spanish neural machine translation enhanced with character and word bitmap fonts. *Machine Translation*, page Accepted for publication.

Marta R. Costa-jussà. 2015. Ongoing study for enhancing chinese-spanish translation with morphology strategies. In *Proc. of the ACL Workshop on Hybrid Approaches to Translation*, Beijing.

Christopher Dyer. 2016. http://code.google.com/p/zhseg/.

Lluís Formiga, Marta R. Costa-jussà, José A. R. Mariño, José B.and Fonollosa, Alberto Barrón-Cedeño, and Lluís Màrquez. 2013. The TALP-UPC phrase-based translation systems for WMT13: System combination with morphology generation, domain adaptation and corpus filtering. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 134–140, Sofia, Bulgaria, August.

Kunihiko Fukushima. 1980. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202.

Jesús Giménez and Lluís Màrquez. 2004. Svmtool: A general pos tagger generator based on support vector machines. In *In Proceedings of the 4th International Conference on Language Resources and Evaluation*. Citeseer.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Sebastien Jean, Orhan Firat, Kyunghun Cho, Roland Memĩsevic, and Yoshua Bengio. 2015. Montreal neural machine translation systems for wmt15. In *Proc. of the 10th Workshop on Statistical Machine Translation*, Lisbon.

Ahmed El Kholy and Nizar Habash. 2012. Rich morphology generation using statistical machine translation. In *Proceedings of the Seventh International Natural Language Generation Conference*, INLG '12, pages 90–94, Stroudsburg, PA, USA. Association for Computational Linguistics.

Philipp Koehn, Franz Joseph Och, and Daniel Marcu. 2003. Statistical Phrase-Based Translation. In *Proc. of the ACL*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicolas Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proc. of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 177–180.

Lluís Padró and Evgeny Stanilovsky. 2012. Freeling 3.0: Towards wider multilinguality. In *Proceedings of the Language Resources and Evaluation Conference (LREC 2012)*, Istanbul, Turkey, May. ELRA.

Michael Paul. 2008. Overview of the iwslt 2008 evaluation campaign. In *Proc. of the International Workshop on Spoken Language Translation*, pages 1–17, Hawaii, USA.

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

Holger Schwenk, Marta R. Costa-jussà, and José A. R. Fonollosa. 2007. Smooth bilingual ngram translation. In *Proc. of the EMNLP*, Prague.

A. Stolcke. 2002. SRILM - an extensible language modeling toolkit. In *7th International Conference on Spoken Language Processing*, Denver.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *CoRR*.

Toshiyuki Takezawa. 2006. Multilingual spoken language corpus development for communicatio n research. In *Chinese Spoken Language Processing, 5th International Symposium, ISCSLP 2006, Singapore, December 13-16, 2006, Proceedings*, pages 781–791.

Kristina Toutanova, Hisami Suzuki, and Achim Ruopp. 2008. Applying morphology generation models to machine translation. In *Proc. of the conference of the Association for Computational Linguistics and Human Language Technology (ACL-HLT)*, pages 514–522, Columbus, Ohio.

Jin Y Yen. 1971. Finding the k shortest loopless paths in a network. *management Science*, 17(11):712–716.

Jiajun Zhand and Chengqing Zong. 2015. Deep neural networks in machine translation: An overview. *IEEE Intelligent Systems*, pages 1541–1672.

Michał Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The united nations parallel corpus v1.0. In *Proceedings of the Tenth International Conference on Language Re sources and Evaluation (LREC 2016)*, may.