Experiments on Document Level Machine Translation

Eva Martínez Garcia
Cristina España–Bonet
Lluís Màrquez

March 2014

Contents

1 Motivation

Most of the current SMT systems work at sentence level. They translate a text assuming that sentences are independent, but, when one looks at a well formed document, it is clear that there exist many inter sentence relations. There is much contextual information that, unfortunately, is lost when translating sentences in an independent way.

We want to improve translation coherence and cohesion using document level information. So, we are interested in develop new strategies to take advantage of context information to achieve our goal. For example, we want to approach this challenge developing postprocesses in order to try to fix a first translation obtained by an SMT system. Also we are interested in taking advantage of the document level translation framework given by the Docent decoder to implement and test some of our ideas.

The analogous problem can be found regarding to automatic MT evaluation metrics because most of them are designed at sentence level so, they do not capture improvements in lexical cohesion and coherence or discourse structure. However, we will left this topic for future work.

2 Document Analysis

First of all, we studied some well formed texts. In particular, we used news from the Newscommentaries2011 corpus. We studied the different translations' errors given by our baseline system[1].

In the analysis we used several automatic tools to obtain a linguistic analysis of the texts. We used some tools inside the Asiya toolkit [GM10]: BIOS [STC05] for Named Entity Recognition, SVMTool tagger for Part-of-Speech tagging [GM04]. We also used the RelaxCor [SPT10] software to obtain a coreference resolution. Also, to obtain another PoS tagging and NER we use the Freeling library [PRAS10].

As a result, we observe the following interesting discourse phenomena:

- Ambiguous words translated in an inconsistent way.
  In this case, we focus on the source words that are translated in different and inconsistent ways. For instance, we can find the word *desk* translated as *escritorio, mesa* and *mostrador* in Spanish, which are not necessarly synonyms. With the assumption of "one sense per discourse", we are interested in treating these kind of words to make our translations more consistent, for instance, using the same translation for all the instances of a source word, going further, using synonyms.
  Although they are not very frequent in a text (maybe there only are about $6-8$ *ambiguous words* in a news document), they have a high impact in the final translation consistency.

- Gender and number disagreements among words in a coreference chain.
  Sometimes it is hard to translate pronouns in a consistent manner through a long text. Furthermore, if we are translating to a morphologically richer language (i.e. when translating from English to Spanish). In particular, the fact that the corefered words in a document agree in gender and number confers a high level of cohesion to the output translation. Hence, we will try to deal with this kind of errors using the information enclosed in the coreference chains.
  We also observe that this kind of mistakes in the gender and number agreement among nouns, adjectives and determiners can be also found at any level of the text.

[1] After setting the interesting errors for us, we start building an SMT baseline system. A Moses decoder trained with the Europarl v7 corpus and tuned with the Newscommentaries2009. We used the English-Spanish language pair.

- Maintaining the discourse structure from source to target.
  A good translation must maintain the discursive structure seen in the source document. For instance, equivalent discourse connectors and the different reasonings in the text.
  In the future work, we are interested in studying the typical errors that appear and think about some strategies to fix or prevent them.

We have implemented a couple of postprocesses that identify and fix the mistakes related with ambiguous words and the gender and number disagreements along all the document.

Regarding to coreference information, we developed a tool that projects the annotated coreferences in the source document in the translated text by means of the alignments given by the Moses decoder. With this projections we want to identify the source coreference chains in the target document. And from there, recognize the gender and number agreement errors. Unfortunately, we only were able of identify two possible fixable examples among the 110 news documents from the test corpus.
Therefore, we decided to move on and use some Wikipedia articles about famous people to try to find more examples where apply our idea, but we did not find any. Hence, we decided to leave this case of study for the future and then propose a new point of view to fix other kind of errors using coreference information or look for other kind of documents where we can find more errors to be able of evaluate our techniques. At the moment, we have left for future work the ideas of keeping the discourse structure and using the information from the coreference chains.

With all this things in mind, we start to develop some techniques to fix these kind of discourse errors.

3 Postprocesses

The first step is trying to identify and fix the described discourse phenomena to be able to improve translation quality.

3.1 Ambiguous Words

We designed a postprocess that identifies at the lemma level those words translated in more than one different way. It uses the alignments given by the decoder and also the PoS tagged source and target documents. It outputs a rewritten source document where the ambiguous words are annotated with translation suggestions to the decoder for a retranslation step.

After identifying the problematic words, we develop two manners of suggesting new translations for them. There is a *restrictive* approach that only suggests the strictly most used translation of the word. In this case, for instance, if *desk* is translated two times as *mesa* and another two times as *mostrador*, our tool will not suggest nothing for the retranslation step. However, when translating *increase*, it appears three times translated as *aumento* and another two as *incremento*; in this case, our tool will tag the source word as follows:

```
<n translation="aumento">increase</n>
```

to force the decoder to retranslate *increase* as *aumento*, which is the most used and, for our heuristic, the most probable to be correct.
But, we can be losing important information when suggesting the new translations. So, we developed a more flexible approach (*probabilistic*) that suggests the translations most used for a source word, giving them a probability. For example, in the previous example with the word *desk*, it will appear tagged as follow:

```
<n translation="mesa||escritorio" prob="0.5||0.5">desk</n>
```

Some details about the postprocess: the software is implemented in Python. It has as input parameters the source text and the translation in their PoS-lemma tagged versions, and also a file with the alignments of the translation. The postprocess goes through all the document and builds a dictionary where stores every "content word"[2] of the input document with the list of the words that have been aligned with it, and a counter with the number of times that

[2]Right now, for us, a content word is a noun, proper noun or and adjective. So, we use the PoS tags to filter out this kind of words.

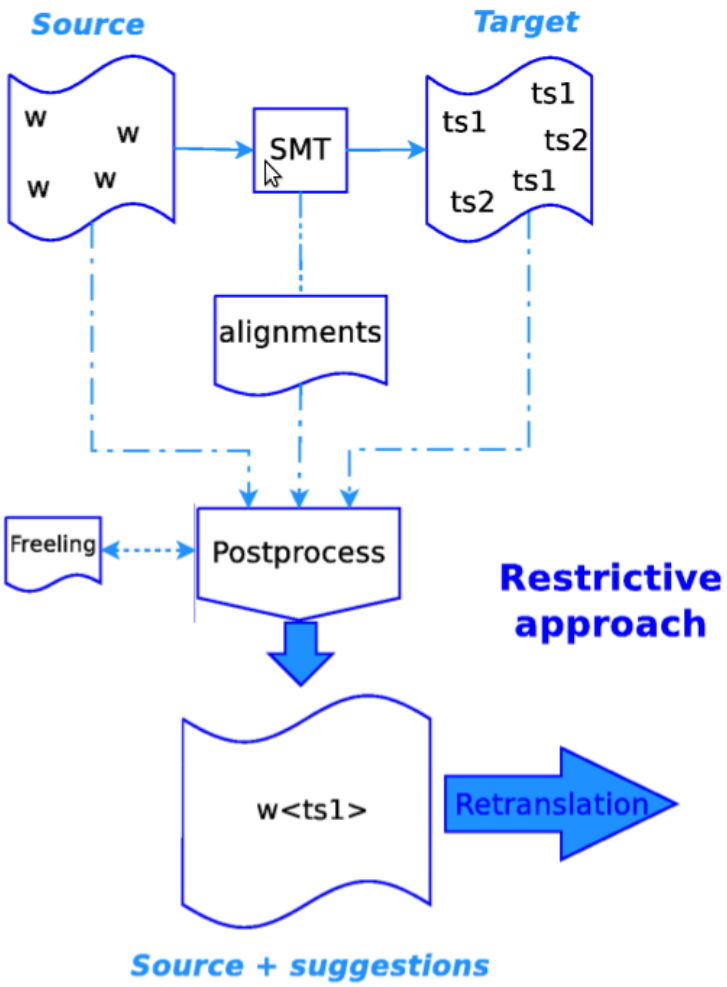

Figure 1: Postprocess for ambiguous words with the restrictive approach that suggests only the most used translation.

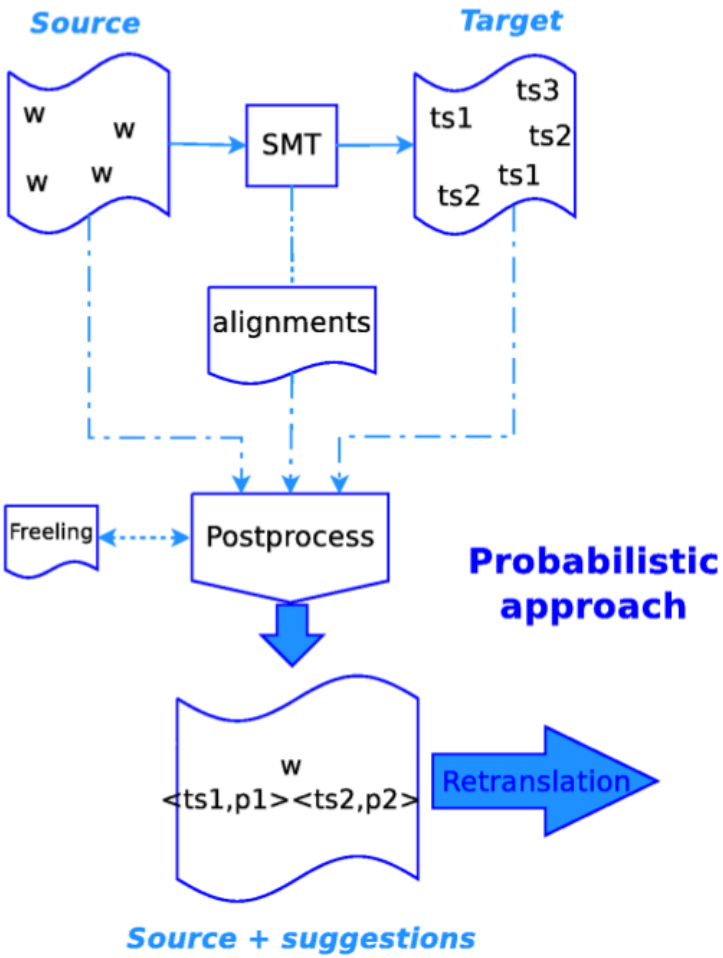

Figure 2: Postprocess for ambiguous words with the probabilistic approach that suggests the most used translations giving them a probability.

an alignment occurs. Hence, when the source text is rewritten, the tool finds the word that want to write in the dictionary and looks for the word(s) to suggest among the aligned ones. Figures 1 and 2 shows the general scheme of how the postprocesses work.

It is important to remember that our postprocesses rewrite the input document with tags that suggest new translations for the inconsistent translated words to the decoder. Once we have rewritten the input text with the suggestions, we need to retranslate it to obtain a new translation to evaluate our system.

document	**#amb.words**	**#changes_rest.**	**#changes_probs.**
news1	10	3	10
news2	6	3	6
news3	14	9	14
news4	14	11	14
news5	24	8	24
news6	29	7	29
news7	15	5	16
news8	1	0	1
news9	3	1	3
news10	6	5	6

Table 1: Number of ambiguous words (*#amb.words*) identified in the news document *newsi*. Also, we show here the number of suggested changes following the restrictive approach (*#changes_rest.*) and the probabilistic approach (*#changes_probs.*)

Manual analysis of the 10 first news in terms of the identified ambiguous words and the introduced changes is shown in Table 1. The number of changes here is the number of ambiguous words with a translation suggestion. This gives us an idea of the impact of our method in the final translation. We have introduced few changes, in the 110 news documents (with 74753 words) we identified 1064 inconsistent translated words. We introduced 476 tags using the restrictive approach and 1064 tags using the probabilistic approach.

Using the usual evaluation metrics we obtained the results shown in Tables 2 and 3. We can say, from the results and the analysis of the introduced changes, that our techniques have a very small impact in the quality of the translation although they are simple approaches. It is remarkable also that the impact is seen using lexical metrics that are not designed to capture this

kind of errors. The automatic metrics that we used are mainly lexical ones and they work using n-gram matches to score the final translations. Since we are changing only about a 1% of the unigrams these metrics are not capturing the changes that our approaches are introducing.

After a manual analysis of the 10 first news documents of our test set, we can say that we are working in a good direction since we are detecting and fixing pretty well the lexical inconsistencies. In a future work we will perform a manual analysis more in depth. However, there is still a lot to do in order to improve the heuristics that we use to choose the suggestions for the retranslation to minimize the introduced noise by our postprocess (use external dictionaries with related topic, refine the frequency counting, use semantics to filter out possible translations that do not fit with the document topic). We realize that our postprocess is introducing noise in the system because the results of the probabilistic approach are worse than the ones form the restrictive one. This is because, sometimes, we are forcing the decoder to use translations that are not correct and that do not appear in the restrictive approach. For example, if we translate *foam*, the restrictive approach will not suggest anything for the retranslation step but, the probabilistic one it is suggested the following:

```
<n translation="poliuretano||espuma" prob="0.5||0.5">
```

We observe that the suggested words here are not always equivalent. This is how we are generating noise in the final translation.

System	**TER**	**BLEU**	**NIST**	**METEOR-ex**	**ROUGE-L**	**ULC**
baseline	55.45	26.73	7.34	27.33	51.51	76.38
restrictive	55.39	26.76	7.34	27.35	51.57	76.50
probabilistic	55.41	26.73	7.34	27.32	51.55	76.44

Table 2: Evaluation of the news as a whole document. The *baseline* is a Moses system trained with the Europarl corpus. *Restrictive* system applies the restrictive postprocessing approach. *Probabilistic* system applies the probabilistic postprocessing approach. Each column corresponds to an automatic evaluation measure.

As a conclusion from these experiments, we can say that studying and treating ambiguous words translated inconsistently has a visible impact in final translations because they are the first things that annoy a human although there are only few examples in a document. Because of that, it will be the first discourse phenomenon that we want to treat from the inside the

System	TER	BLEU	NIST	METEOR-ex	ROUGE-L
baseline_news1	68.87	10.79	3.83	19.66	42.75
restrictive_news1	68.87	10.80	3.83	19.64	42.52
probabilistic_news1	69.09	10.98	3.82	19.53	42.44
baseline_news2	63.69	20.73	4.27	24.57	47.21
restrictive_news2	63.69	20.73	4.27	24.57	47.21
probabilistic_news2	63.69	20.73	4.27	24.57	47.21
baseline_news3	66.79	14.15	4.19	21.53	42.34
restrictive_news3	66.72	14.16	4.19	21.54	42.39
probabilistic_news3	65.97	14.23	4.22	21.69	42.58
baseline_news4	67.55	18.69	4.03	21.81	43.65
restrictive_news4	67.40	18.93	4.06	22.07	44.11
probabilistic_news4	67.11	18.94	4.07	22.14	44.31
baseline_news5	69.15	13.74	3.73	20.06	39.99
restrictive_news5	69.15	13.74	3.73	20.06	39.99
probabilistic_news5	69.06	13.75	3.73	20.04	40.09

Table 3: Examples of evaluations of single documents. Systems as in Table 2 but for a particular document news$i$.

Docent decoder. First we want to design a *feature function* able to evaluate the ambiguity of a document regarding to the number of the ambiguous words. Afterwards, we find interesting to develop a change operation inside the decoder.

3.2 Gender and number

We go back on the idea of dealing with disagreements in gender and number along the document. We designed a postprocess that checks the agreement among nouns, determiners and adjectives.

Figure 3 shows a scheme of the postprocess. As in the previous case it uses as input the source document, the translation (with their PoS and lemmas tags) and the alignments. Then, for every word in the input source document, if it is a noun, the postprocess checks if the related determiners have the same gender and number by means of the PoS tags. If there is any disagreement, it uses the dictionary functions from the Freeling library [PRAS10] to generate the correct form of the determiners. For instance, if we have in the translation the following chain: *el casa*, the postprocess sees that there is a disagreement between the gender of the determiner *el* (masculine) and the gender of the noun *casa* (feminine), and generates the singular feminine form of the determiner and rewrites the chain as ***la*** *casa*.
The postprocess also makes the checking among nouns and their related adjectives, making them to agree with the gender and number of the noun.

Once again, we test our implementation over the Newscommentaries2011 corpus. We did a manual study of the number of introduced changes in order to evaluate the impact of this simple technique. The results are shown in Table 4.

The results obtained applying this technique are shown in Tables 5 and 6. Once again, we observe few improvements, but we cannot forget that we are making only few changes (< 1000 changed words in a corpus of 75000 words). The following step will be studying the agreement along coreference chains and also look at the agreement among verbs and the subjects.

As a conclusion from these experiments, we can say that the phenomena of gender and number disagreement is more frequent that we expected. However, we are again introducing only few changes that the automatic lexical metrics are not able to reflect in the scores of the translations. It will be necessary a manual analysis and evaluation in depth. We will also be interested in including this information inside a feature function in order to guide the decoder to choose and/or produce more coherent translations.

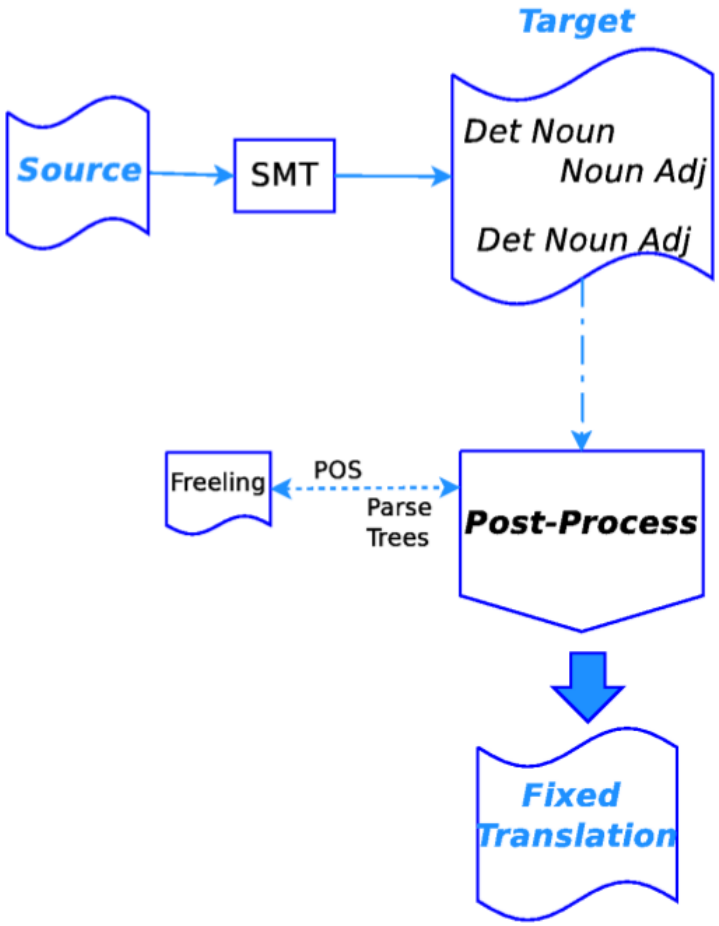

Figure 3: Postprocess that deals with disagreements in gender and number among nouns, determiners and related adjectives through all the document. Finally, it outputs a fixed translation.

document	**#changes_dets**	**#changes_dets+adj**
news1	9	12
news2	3	7
news3	4	10
news4	2	11
news5	14	24
news6	7	30
news7	1	6
news8	4	17
news9	0	0
news10	1	2

Table 4: Changes when checking gender and number agreements in several documents (*newsi*). In particular, checking agreement among nouns and determiners (*#changes_dets*) and among nouns, determiners and adjectives (*#changes_dets+adj*)

.

System	**TER**	**BLEU**	**NIST**	**METEOR-ex**	**ROUGE-L**	**ULC**
baseline	55.45	26.73	7.34	27.33	51.51	76.33
agreement_nn+dets	55.40	26.76	7.34	27.37	51.61	76.49
agreement_nn+dets+adj	55.38	26.69	7.34	27.34	51.56	76.42

Table 5: Examples of evaluations of the newswire documents as a whole set. We apply each metric to the full test set. The *baseline* is a Moses system trained with the Europarl corpus. *Agreement_nn+dets* system checks only the agreement between nouns ant their determiners. *Agreement_nn+dets+adj* system checks the agreement among nouns and their determiners and also their related adjectives.

System	TER	BLEU	NIST	METEOR-ex	ROUGE-L
baseline_news1	68.87	10.79	38.26	19.66	42.75
agreement_nn+dets_news1	68.21	11.87	38.78	19.97	43.02
agreement_nn+dets+adj_news1	67.99	11.88	38.90	20.02	43.10
baseline_news2	63.69	20.73	42.68	24.57	47.21
agreement_nn+dets_news2	63.50	20.79	42.86	24.69	47.59
agreement_nn+dets+adj_news2	63.50	20.72	42.60	24.49	47.52
baseline_news3	66.79	14.15	41.89	21.53	42.34
agreement_nn+dets_news3	66.87	14.16	41.83	21.50	42.25
agreement_nn+dets+adj_news3	66.79	14.16	41.79	21.48	42.36
baseline_news4	67.55	18.69	40.30	21.81	43.65
agreement_nn+dets_news4	67.40	18.73	40.47	21.89	43.77
agreement_nn+dets+adj_news4	67.40	18.83	40.51	21.91	44.02
baseline_news5	69.15	13.74	37.29	20.06	39.99
agreement_nn+dets_news5	68.79	13.85	37.69	20.32	40.29
agreement_nn+dets+adj_news5	68.61	13.92	37.96	20.43	40.14

Table 6: Examples of evaluations of single documents after using the post–process to correct gender and number disagreements. As in Table 5 but for individual news documents.

4 Future work: using Docent

We are used to deal with MT sistems that make translations sentence by sentence or phrase by phrase. Also, they assume independence among sentences. However, the Docent decoder [HSTN13, HNT12] considers an entire document at translation time. It applies changes and makes evaluations of possible translations at document level to finally obtain a final translation. So, this decoder takes into account contextual information to translate a document.

4.1 Docent: a document level decoder

Docent is a document level oriented decoder. It is build on top of a Moses system [KHB+07] (which uses to get a first translation to start the process) but it applies changes and evaluates the different translation candidates looking at document level features.
In every step of the translation process, Docent applies a change operation over the document (in particular, to the current translation state of the document) and then evaluates the possible translations that are in the neighbourhood of candidates and give them a score. Finally, the decoder choses the best translation according to the calculated scores for the entire document.

Figure 4 shows a simplified scheme of the Docent decoder. There are

two different parts, the *sampler* and the *scorer*. The *sampler* has inside a little modules called *change operations*. These operations describe different changes that can be applied to a translation state (change phrases, reorder words, etc.). In particular, these operations do not know anything about the context information or the discourse of the text, they simply apply several techniques to generate new translation states.
On the other hand, the *scorer* has modules called *feature functions* that calculate a score for a translation state. These functions take into account different characteristics to promote those possible translations that are better for us, for instance, there can be a feature function that help the system to filter out those translations with a high level of ambiguous words translated inconsistently.

The default operations implemented in Docent are: *change-phrase-translation* (changes a translation of a phrase for a random translation from the phrase table that overlaps the same words as the original), *resegment* (is the more complex operation. Allows the decoder to modify the segmentation of the phrase in the source) and *swap-phrases* (affects the word order in the output without changing the phrase translation). After applying these operations, Docent calculates the score of the resulting translations. It uses the feature functions to do that. Every feature fuction evaluates a different desireable characteristic in the final translation.

We want to take advantage of the Docent framework to develop techniques that allow us to improve translation quality at document level. So, we will start working with Docent designing new feature functions that take into account the phenomena that we have described before. Later on, we want to develop new change operations (at least one) that are able to minimize the appearance of these phenomena in the neighbourhood of translation candidates.

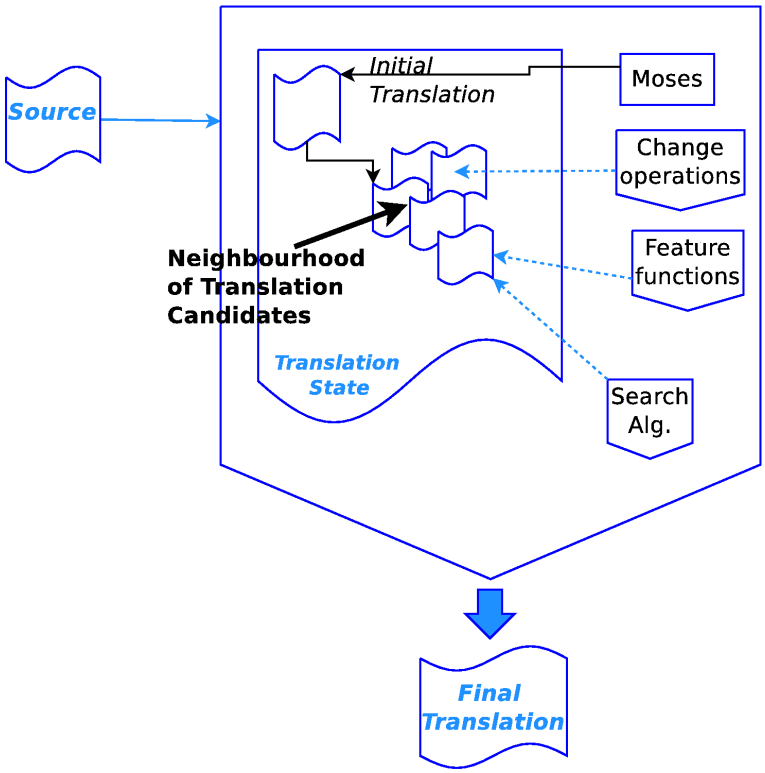

Figure 4: Structure of Docent system.

4.2 Feature Functions

Docent decoder works treating a document as a whole at translation time. However, its feature functions can take into account several characteristics at every level of the document (sentences, phrases and words) to assign a score to the translation state.

We will start designing a quite simple version of a function that captures the ambiguity of a translation looking at the amount of ambiguous words translated in an inconsistent way. This function will assign a score given by the proportion of ambiguous words in the document. For instance:

$$f_{amb}(x) = \frac{\#ambiguos\ words}{\#words}$$

We plan to implement this function using the alignments of every translation to identify the inconsistently translated words.

The following step here will be to refine this function making pondering the score taking into account the number of appearances of every translation option. In that way, we will be able to distinguish among words more or less ambiguous (i.e., *desk* translated two times as *escritorio*, two times as *mesa* and three times as *mostrador* should be more ambiguous than *house* translated five times as *cámara* and three times as *casa*).

We also want to develop new functions that focus on the different interesting discourse phenomena, like disagreements in gender and number or the use of connectors, discourse markers and looking at coreference information.

References

[GM04] Jesús Giménez and Lluís Màrquez. Svmtool: A general pos tagger generator based on support vector machines. In *Proceedings of the 4th LREC*, Lisbon, Portugal, 2004.

[GM10] J. Giménez and L. Màrquez. Asiya: An open toolkit for automatic machine translation (meta-)evaluation. In *Prague Bulletin of Mathematical Linguistics, 94*, pages 77–86, 2010.

[HNT12] C. Hardmeier, J. Nivre, and J. Tiedemann. Document-wide decoding for phrase-based statistical machine translation. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1179–1190, Jeju Island, Korea, 2012.

[HSTN13] C. Hardmeier, S. Stymne, J. Tiedemann, and J. Nivre. Docent: A document-level decoder for phrase-based statistical machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics (ACL)*, pages 193–198, Sofia, Bulgaria, 2013.

[KHB+07] 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. Moses: open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting on Association for Computational Linguistics*, pages 177–180, Prague, Czech Republic, 2007.

[PRAS10] Lluís Padró, Samuel Reese, Eneko Agirre, and Aitor Soroa. Semantic services in freeling 2.1: Wordnet and ukb. In Pushpak Bhattacharyya, Christiane Fellbaum, and Piek Vossen, editors, *Principles, Construction, and Application of Multilingual Wordnets*, pages 99–105, Mumbai, India, February 2010. Global Wordnet Conference 2010, Narosa Publishing House.

[SPT10] Emili Sapena, Lluís Padró, and Jordi Turmo. A global relaxation labeling approach to coreference resolution. In *Proceedings of 23rd International Conference on Computational Linguistics, COLING*, Beijing, China, August 2010.

[STC05] M. Surdeanu, J. Turmo, and E. Comelles. Named entity recognition from spontaneous open-domain speech. In *Proceedings of*

the 9th International Conference on Speech Communication and Technology (Interspeech), 2005.