Linguistic Categorisation in Machine Translation using Stochastic Finite State Transducers[1]

Jorge González and Francisco Casacuberta
Departamento de Sistemas Informáticos y Computación
Instituto Tecnológico de Informática
Universidad Politécnica de Valencia
{jgonzalez, fcn}@dsic.upv.es

Abstract

In the last years, statistical machine translation has already demonstrated its usefulness within a wide variety of translation applications. In particular, finite state models are always an interesting framework because there are well-known efficient algorithms for their representation and manipulation. Nevertheless, statistical approaches have rarely been performed taking into account the linguistic nature of the translation problem. This document describes some methodological aspects of building category-based finite state transducers that are able to consider a set of linguistic features in order to produce the most linguistically appropriate hypotheses.

1 Introduction

Machine Translation (MT) is a consolidated area of research in computational linguistics which investigates the use of computer software to translate text or speech from one natural language to another. The goal of MT is very ambitious because it would involve a reduction of the linguistic barriers in human communication.

Despite their initial relative success, rule-based systems were quickly challenged by their rival inductive approaches, which adopt some pattern recognition techniques to learn the models. *Statistical* machine translation represents an interesting framework because the translation software is language-independent, that is, different MT systems are built if different parallel corpora are supplied.

Given a source sentence $\mathbf{s}_1^J = \mathbf{s}_1 \ldots \mathbf{s}_J$, the goal of statistical machine translation is to find a target sentence $\hat{\mathbf{t}}_1^I = \mathbf{t}_1 \ldots \mathbf{t}_I$, among all the possible target strings $\mathbf{t}_1^I$, that maximises the posterior probability of $\mathbf{t}_1^I$ given $\mathbf{s}_1^J$:

$$\hat{\mathbf{t}}_1^I = \underset{\mathrm{t}_1^I}{\operatorname{argmax}} \Pr(\mathbf{t}_1^I | \mathbf{s}_1^J) \qquad (1)$$

Since $\Pr(\mathbf{s}_1^J)$ is independent of $\mathbf{t}_1^I$, the equation (1) can be rewritten to (2), using a joint probability distribution that is modelled by means of stochastic finite state transducers:

$$\hat{\mathbf{t}}_1^I = \underset{\mathrm{t}_1^I}{\operatorname{argmax}} \Pr(\mathbf{s}_1^J, \mathbf{t}_1^I) \qquad (2)$$

Despite the linguistic nature of languages has been traditionally ignored in statistical machine translation, there is some recent related work that tries to incorporate some linguistic knowledge into a statistical framework (Niessen, 2004; Gispert, 2006; Koehn, 2006).

The organization of this paper is as follows: next section presents the statistical framework; section 3 describes the methodological aspects of building a category-based system, where training and decoding steps are

[1] This work was supported by the EC (FEDER) and the Spanish projects TIN2006-15694-C02-01 and the Consolider Ingenio 2010 CSD2007-00018

explained in depth; the experimental setup and results are shown in section 4; finally, conclusions are briefly summed up at section 5.

2 Statistical framework

Machine translation can be seen as a process of pattern recognition, where objects to be tested are sentences from a source language. These sentences should be coded in a process of feature extraction in order to be classified or described by a previously estimated model.

On the one hand, geometric feature extraction defines a real object **s** as a feature vector where every observed feature is measured on **s** and then annotated to the right position. On the other hand, syntactic feature extraction establishes a structural description of **s**, according to some structure-based instructions.

Given that a text sentence **s** represents a structural description, i.e. a string of symbols, these word sequences have been traditionally employed in the field of computational linguistics as a result of a feature extraction process.

However, nobody ignores that the linguistic nature of languages could be statistically exploited in order to obtain some better models. In such a line, every word in a sentence is expanded into a tuple of three different pieces of information: on the one hand, the written word itself, also known as surface form; on the other hand, its base form, also referred in the literature as lemma; finally, a linguistic feature vector reports information about its lexical category together with a set of linguistic properties, such as gender, number, etc. In this way, a traditional definition of $\mathbf{s} = \mathbf{s}_1 \ldots \mathbf{s}_J$ would be replaced by an extended string $\mathbf{s} = (\mathbf{s}_1, m_1, u_1) \ldots (\mathbf{s}_J, m_J, u_J)$, where m_j stands for the lemma of word $\mathbf{s}_j$, and u_j stands for its linguistic feature vector.

Given that a lemma can be seen as a linguistic cluster, where words sharing the same lemma are classified into the same cluster, the vocabulary can be significantly reduced by changing the words to their lemmas during the estimation of the joint probability model.

Let $\mathbf{s} = (\mathbf{s}_1^J, m_1^J, u_1^J)$ and $\mathbf{t} = (\mathbf{t}_1^I, n_1^I, v_1^I)$ be a source and a target sentence respectively, equation 2 can be tackled through a categorisation scheme:

$$\begin{aligned}\Pr(\mathbf{s}_1^J, \mathbf{t}_1^I) \;=\; & \Pr(m_1^J, n_1^I) \cdot \Pr(u_1^J | m_1^J, n_1^I) \cdot \\ & \Pr(v_1^I | m_1^J, n_1^I, u_1^J) \cdot \\ & \Pr(\mathbf{s}_1^J | m_1^J, n_1^I, u_1^J, v_1^I) \cdot \\ & \Pr(\mathbf{t}_1^I | m_1^J, n_1^I, u_1^J, v_1^I, \mathbf{s}_1^J)\end{aligned}$$

which, under certain assumptions, turns to:

$$\begin{aligned}\Pr(\mathbf{s}_1^J, \mathbf{t}_1^I) \;\approx\; & \Pr(m_1^J, n_1^I) \cdot \Pr(u_1^J | m_1^J) \cdot \\ & \Pr(v_1^I | n_1^I) \cdot \Pr(\mathbf{s}_1^J | m_1^J, u_1^J) \cdot \\ & \Pr(\mathbf{t}_1^I | n_1^I, u_1^J, v_1^I)\end{aligned}$$

Lemma-based joint probability distributions $\Pr(m_1^J, n_1^I)$ can be modelled by stochastic finite state transducers, whereas specialised stochastic dictionaries can be estimated to model uncategorising lemma-to-word transformations $n_1^I \rightarrow \mathbf{t}_1^I$, according to a given source feature vector u_1^J, assuming that $\Pr(\mathbf{t}_1^I | n_1^I, u_1^J, v_1^I)$ is also independent of v_1^I. This behaviour is based on a Spanish↔Catalan machine translation system (González, 2006) which assumes that linguistic information is transferred from input to output, remaining unaltered in most cases.

The equation (2) will then be expressed as:

$$\begin{aligned}\hat{n}_1^I &= \operatorname*{argmax}_{n_1^I} \Pr(m_1^J, n_1^I) \\ \hat{\mathbf{t}}_1^I &= \operatorname*{argmax}_{\mathbf{t}_1^I} \Pr(\mathbf{t}_1^I | \hat{n}_1^I, u_1^J)\end{aligned} \tag{3}$$

The search must be constrained in order to perform first a lemma transduction operation, that is, translating from source to target lemmas, then turning lemmas into words, through their corresponding feature vectors.

Specialised stochastic dictionaries can be estimated following the maximum likelihood approach in order to compute $\Pr(\mathbf{t}_1^I | \hat{n}_1^I, u_1^J)$. The specialisation criteria can be seen from two equivalent points of view: on the one hand, a stochastic dictionary can be trained for every different target lemma, thus every entry informs about how a feature vector can be translated into a target word; or, maybe more intuitively, training a lemma-to-word stochastic dictionary per each feature vector. The calculation of $\Pr(\mathbf{t}_1^I|\hat{n}_1^I, u_1^J)$ is carried out by means of the contribution of all the individual translation probabilities, that is:

$$\Pr(\mathbf{t}_1^I|\hat{n}_1^I, u_1^J) \approx \prod_{i=1}^{I} \Pr(\mathbf{t}_i|\hat{n}_i, u_{\alpha_i})$$

Formally, an alignment function α is a mapping $\alpha : i \to j$ that assigns a source position j to a target position i, $\alpha_i = j$. Alignments are used as hidden variables in statistical machine translation models such as IBM models (Brown, 1990) or hidden Markov models (Zens, 2002). Therefore, target lemmas being generated are able to know which source position was responsible for their occurrence.

3 Probabilistic models

A weighted finite-state automaton is a tuple $\mathcal{A} = (\Gamma, Q, i, f, P)$, where Γ is an alphabet of symbols, Q is a finite set of states, functions $i : Q \to \mathbb{R}$ and $f : Q \to \mathbb{R}$ give a weight to the possibility of each state to be initial or final, respectively, and partial function $P : Q \times \{\Gamma \cup \{\lambda\}\} \times Q \to \mathbb{R}$ defines a set of transitions between pairs of states in such a way that each transition is labelled with a symbol from Γ or the empty string λ, and is assigned a weight.

A weighted finite-state transducer (Mohri, 2002; Kumar, 2006) is defined similarly to a weighted finite-state automaton, with the difference that transitions between states are labelled with pairs of symbols that belong to the cartesian product of two different (input and output) alphabets, $\{\Sigma \cup \{\lambda\}\} \times \{\Delta \cup \{\lambda\}\}$.

When weights are probabilities, and under certain conditions, a weighted finite-state model can define a distribution of probabilities on the free monoid. In that case it is called a stochastic finite-state model. Then, given some input/output strings $\mathbf{s}_1^J$ and $\mathbf{t}_1^I$, a stochastic finite-state transducer is able to associate a probability $\Pr(\mathbf{s}_1^J, \mathbf{t}_1^I)$ to them.

3.1 Inference of stochastic transducers

The GIATI paradigm (Casacuberta, 2005) has been revealed as an interesting approach to infer stochastic finite-state transducers through the modelling of languages. Rather than learning translations, GIATI first converts every pair of parallel sentences from the training corpus into only one string to, after all is done, infer a language model from.

More concretely, given a parallel corpus consisting of a finite sample C of string pairs: first, each training pair $(\bar{x}, \bar{y}) \in \Sigma^\star \times \Delta^\star$ is transformed into a string $\bar{z} \in \Gamma^\star$ from an extended alphabet, yielding a string corpus S; then, a stochastic finite-state automaton $\mathcal{A}$ is inferred from S; finally, transition labels in $\mathcal{A}$ are turned back into pairs of strings of source/target symbols in $\Sigma^\star \times \Delta^\star$, thus converting the automaton $\mathcal{A}$ into a transducer $\mathcal{T}$.

The first transformation is modelled by some labelling function $\mathcal{L} : \Sigma^\star \times \Delta^\star \to \Gamma^\star$, whereas the last transformation is defined by an inverse labelling function $\Lambda(\cdot)$, such that $\Lambda(\mathcal{L}(C)) = C$. Building a corpus of extended symbols from the original bilingual corpus allows for the use of many useful algorithms for learning stochastic finite-state automata (or equivalent models) that have been proposed in the literature about grammatical inference.

Every extended symbol from Γ has to condense somehow the meaningful relationship that exists between the words in the input and output sentences. Discovering these relations is a problem that has been throughly studied in statistical machine translation and has well-established techniques for dealing with it. The concept of statistical alignment formalises this problem. Whether this function is constrained to a one-to-one, a one-to-many or a many-to-many correspondence depends on the particular assumptions that we make. Constraining the alignment function simplifies the learning procedure but reduces the expressiveness of the model. The available algorithms try to find a trade-off between complexity and expressiveness.

One-to-one and one-to-many alignment functions would enable models to adopt the categorisation scheme presented here because they allow for alignments where one target position is aligned to only one source position.

One-to-one models do not seem a very appropriate approach provided that they would require that source-target aligned sentences had exactly the same number of words. Nevertheless, one-to-many alignment models are a current reference in machine translation research community by means of their well-known IBM models (Brown, 1990).

A smoothed n-gram model may be inferred from the string corpus previously generated. Such a model can be expressed in terms of a weighted finite-state automaton. Since every transition consumes only one symbol, and given that all those extended symbols are composed of exactly one source element, the inverse labelling function can be straightforwardly applied. This way, transition labels are turned back into pairs of source and target items, thus becoming a stochastic transducer.

3.1.1 Alignment models

The conversion of every pair of parallel sequences into an extended symbols string follows this algorithm: for each target item from left to right, merge it with its corresponding source element iff the alignment does not cross over any other alignment, in which case it is delayed and attached to the last implied source item. Spurious source and target elements are placed at their right position, given that a monotonous order is always demanded. This procedure ensures that every extended symbol is composed of one and only one source symbol, optionally followed by an arbitrary number of target symbols. For a more detailed description about the labelling function, see (Casacuberta, 2005).

The implementation of the categorisation scheme will require increasing the information to be included in every compound symbol. More concretely, all the target lemmas being produced by the model need to report which relative source position they are coming from.

Figure 1 displays the two situations which the labelling function may be involved with.

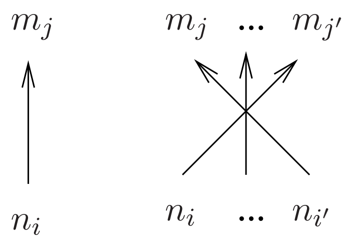

Figure 1: Two types of alignments

Whereas the first example (namely, the relation $n_i \rightarrow m_j$) is undoubtedly easy to solve, the second one implies a little more of work. One-to-one relationships clearly establish that n_i is aligned to the current source symbol being analysed m_j. This is denoted as a relative movement of 0, as it can be seen in figure 2.

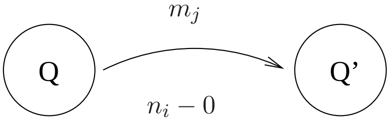

Figure 2: One-to-one compound symbols

On the other hand, crossing alignments would imply delaying the output of $\{n_i \ldots\}$ until $m_{j'}$ is being parsed, then producing the full target segment $n_i \ldots n_{i'}$. Therefore, every lemma being generated may not be aligned with its corresponding input symbol as before, but with some previously parsed one instead.

As a consequence, target lemmas are annotated together with their relative distance to the source lemma which they were aligned to. Spurious elements do not need such annotation because of their own spontaneous generation, which is independent of any particular source item. In figure 1, n_i is aligned to the current source element $m_{j'}$, thus indicated as a 0 relative movement. However, the emission of $n_{i'}$ will be delayed, then moving it further away from its aligned input item m_j. This relative distance is then annotated next to the output symbol $n_{i'}$ as a reminder to allow for a posterior backtracking performance. The result of such a labelling algorithm can be seen over the final transducer, as figure 3 shows.

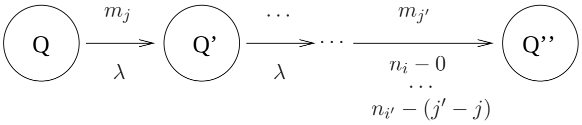

Figure 3: One-to-many compound symbols

Note that the relative distance for $n_{i'}$ is computed as the subtraction of the input position aligned to, j, from the current one, j'.

3.2 The search problem

Word-to-word translation as in equation (2), or lemma-to-lemma translation as in the first equation of (3), are expressions of the MT problem in terms of a finite state model that is able to compute a joint probability. Given that only the input sentence is known, the model has to be parsed, taking into account all the outputs that are compatible with the input. The best target hypothesis would be that one which corresponds to a path through the transduction model that, with the highest probability, accepts the input sequence as part of the input language of the transducer.

Although the navigation through the model is constrained by the input sequence, the search space can be extremely large. As a consequence, only those partial hypotheses with the highest scores are being considered as possible candidates to become the solution. This search process is very efficiently carried out by the well known Viterbi algorithm.

3.3 Stochastic dictionaries

A weighted dictionary is a table $(a, b, W(a, b))$ containing a set of translation pairs together with a numerical indicator for their reliability. If $W(a, b) = \Pr(a|b)$ and $\forall y \sum_x \Pr(x|y) = 1$, then it can be called a stochastic dictionary.

Once a lemmatised source sentence has been analysed by the transduction model, output is expressed as a sequence of target lemmas. They can be turned into their corresponding surface forms by means of specialised stochastic dictionaries that take into account the linguistic information of the source elements which they are attached to.

Following the maximum likelihood approach, a stochastic dictionary can be estimated by counting the absolute frequencies of the observed events, properly normalised:

$$\Pr(t_i|n_i) = \frac{F(t_i, n_i)}{\sum_x F(x, n_i)}$$

These dictionaries can be learnt by means of two different estimation methods: one considers only a monolingual target corpus, thus learning conversions through their own target linguistic information; and another one that takes into account the statistical alignments over a bilingual corpus in order to train lemma-word transformations according to their corresponding source feature vectors. In this case, the alignments that are needed for learning the stochastic lemma-based transducers are also adequate for the extraction of the lemma-to-word relative frequencies. An outline of this method is depicted in figure 4.

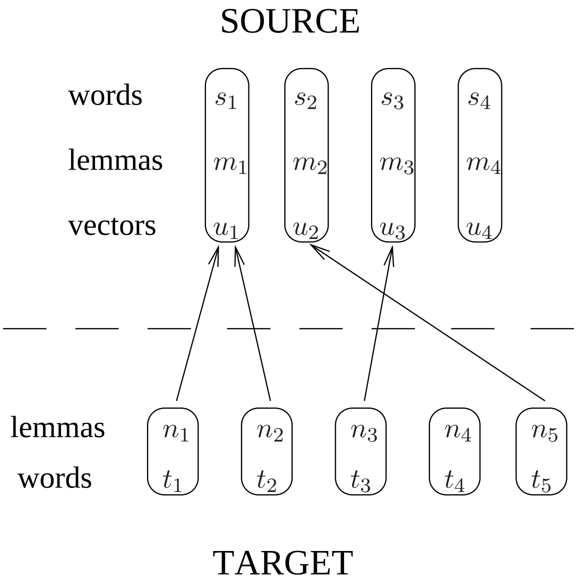

Figure 4: Using the source vectors for a bilingual estimation of lemma-to-word dictionaries

3.4 On-the-fly integrated architecture

The equations in (3) represent the search strategy in order to translate a test sentence from a source language to a target language. According to these equations, translation is carried out in two separate steps: first, source lemmas are transformed into target lemmas through a finite state approach, then lemmas are turned into words by means of specialised, linguistic-based stochastic dictionaries.

However, this two step procedure can be integrated into only one process, thus merging the lemma-word conversions into the parsing algorithm of the lemmatised input sentences.

Let j be a current analysis position of the input sequence m_1^J, and let n_i be a target lemma being produced during the parsing of m_j. Given that a lemma-word translation probability $\Pr(\mathbf{t}_i|n_i)$ has been assumed to also (and only) depend on the source feature vector u_{α_i} which n_i has been aligned to, and since α_i is always guaranteed to be a position $0 \leq \alpha_i \leq j$ that has already been analysed, then $\Pr(\mathbf{t}_i|n_i, u_{\alpha_i})$ can be applied in order to turn a target lemma n_i into a target word $\mathbf{t}_i$.

Thanks to including the alignment information in between the output symbols, it is possible to know for each lemma being generated which input position it has been connected to.

As a result, every target lemma being produced as part of a partial output hypothesis may be converted and stored as a target word, without the need for waiting for the best output hypothesis $\hat{n}_1^I$ to be completely generated.

Once the input sequence m_1^J has been fully parsed through the finite state model, a final surface form $\hat{\mathbf{t}}_1^I$ has been produced on the fly.

4 Experiments

A set of preliminary experiments were carried out in order to test the viability of our integrated category-based translation approach.

Two tasks of very different difficulty degrees were employed for the design of the experimental setup. The EuTrans task is defined on the restricted domain of sentences that a tourist traveller would say at a hotel's desk. It is artificially generated from a set of schemas of sentences. The characteristics of the EuTrans corpus can be seen in table 1. Spanish to English translation was carried out over this low-perplexity task.

On the other hand, this approach has been also applied to a Portuguese–Spanish section of the EuroParl corpus. The EuroParl corpus is built on the proceedings of the European Parliament, which are published on its web and are freely available. Because of its nature, this corpus has a large variability and complexity, since the translations into the different official languages are performed by groups of human translators. The fact that not all translators agree in their translating criteria implies that a given source sentence can be translated in various different ways throughout the corpus. Since the proceedings are not available in every language as a whole, a different subset of the corpus is extracted for every different language pair, thus evolving into somewhat different corpora for each pair. The corpus characteristics can be seen in table 2.

Table 1: EuTrans corpus characteristics

EuTrans		Spanish	English
Training	Sentences	10.000	
	Run. words	97.1K	99.3K
	Vocabulary	686	513
Closed test	Sentences	2.996	
	Perplexity	4.9	3.6
Open test	Sentences	3.000	
	Perplexity	4.9	3.6

Table 2: Characteristics of pt–es EuroParl

EuroParl		Portuguese	Spanish
Training	Sentences	915.570	
	Run. words	23.76M	23.95M
	Vocabulary	141.6K	140.4K
Sub-train	Sentences	50.000	
	Run. words	1.3M	1.3M
	Vocabulary	37.3K	37.6K
Test	Sentences	1.000	
	Train pp.	71.9	66.2
	Sub-train pp.	121.3	103.5

EuTrans lemmatisation and linguistic labelling were carried out through the FreeLing toolkit (Carreras, 2004), whereas SisHiTra (González, 2006) was employed to analyse the Spanish sentences from EuroParl. Portuguese lemmas and feature vectors were provided by the Spoken Language Systems Laboratory from the Instituto de Engenharia de Sistemas e Computadores I+D in Lisbon. Both EuTrans and EuroParl corpora were aligned at word level by means of the toolkit GIZA++.

Several tokenisation options were tested to establish a starting point where the categorisation scheme proposed here could be applied.

4.1 Evaluation metrics

The results were obtained by using the following evaluation measures:

BLEU *(Bilingual Evaluation Understudy) score*: This indicator computes the precision of unigrams, bigrams, trigrams, and tetragrams with respect to a set of reference translations, with a penalty for too short sentences. BLEU measures accuracy, not error rate.

WER *(Word Error Rate)*: The WER criterion calculates the minimum number of editions (substitutions, insertions or deletions) needed to convert the system hypothesis into the sentence considered ground truth. Because of its nature, this measure is a pessimistic one.

4.2 Translation results

EuTrans is a very artificial translation task which is frequently used for debugging purposes. New approaches to statistical machine translation are first tested on such a toy task in order to establish some behaviour criteria. The EuTrans results are reported in table 3.

Table 3: EuTrans results

	Vocab.		Pp.		Metrics	
EuTrans	In	Out	In	Out	WER	BLEU
Baseline	686	513	4.9	3.6	8.3	88.0
Tokenisation	624	513	5.2	3.6	**8.1**	88.0
Categorisation	476	503	4.6	3.6	11.8	82.0
Monolingual					22.0	64.3
Bilingual					13.1	78.9

As it can be seen, our linguistic categorisation approach is not worth the trouble for EuTrans. Tokenisation techniques do perform a slight improvement on word error rate, but lemmatisation make results get worse. Whereas the results from “Categorisation” lines represent a comparison with a predefined lemmatised reference, thus evaluating somehow the effect of the lemma transduction model, “Monolingual” and “Bilingual” lines refer to the overall process of translation, according to the way specialised stochastic lemma-to-word dictionaries were learnt. Therefore, the “Categorisation” error rates are always a lower limit of the overall system. It can also be appreciated that there is a significative difference between using a monolingual or a bilingual lemma-to-word approach.

On the other hand, EuroParl is a more complex task which is reflected through its vocabulary and perplexity figures (see table 2). Due to technical issues, experiments were carried out by using only a subset of the training corpus, which is composed of 50.000 sentences. Lemmatisation can reduce vocabularies about 50%, thus causing perplexities to significatively fall as well, as table 4 shows.

Table 4: EuroParl vocabulary and perplexity

	Vocab.		Pp.	
EuroParl	In	Out	In	Out
Baseline	37.3K	37.6K	121.3	103.5
Tokenisation	37.3K	37.5K	121.3	120.9
Categorisation	18.3K	19.3K	91.1	91.1

The EuroParl results are reported in table 5.

Table 5: EuroParl results

	Metrics		Model size	
EuroParl	WER	BLEU	States	Arcs
Baseline	67.8	19.8	205K	1.06M
Tokenisation	**65.7**	**20.0**	200K	1.04M
Categorisation	**61.3**	**23.0**	166K	925K
Monolingual	81.0	3.0	38K	
Bilingual	**63.2**	**21.4**	94K	

In this case, using morphologically annotated corpora helps to the translation process. As well as tokenisation, categorisation also allows for a better modelling of transference relations between source and target languages. The sizes of the models are also significatively reduced, which means not only a memory saving, but also accelerating the decoding time.

Globally, if a bilingual approach is followed to estimate the lemma-word dictionaries, thus using the *source* linguistic feature vectors to specialise them, then the methodology presented here outperforms the baseline system.

Again, monolingual estimation of dictionaries does not perform well and table 6 can show the reasons for such a so different behaviour.

Table 6: Analysis of lemma-word conversions. An impact is defined as a successful search over the lemma-word dictionaries. If the search fails, then lemmas are left unchanged.

Training		EuTrans	EuroParl
	Spurious	3.6%	8.1%
Monolingual	Impacts	11.3%	0%
	Fails	85.1%	91.9%
Bilingual	Impacts	93.1%	88.5%
	Fails	3.3%	3.4%

From table 6, it seems quite clear why monolingual training is doing worse. Impacts and fails are oppositely distributed with respect to the ones from a bilingual training. Whereas a bilingual training reflects an approximate 90% of impacts, a monolingual training associates this percentage to fails. If most lemmas remain unchangeable, then the evaluation results from tables 3 and 5 can be explained, since the lemma-based hypotheses are being compared to word-based references.

Massive fails for a monolingual training are caused by a mismatch between source and target feature vectors. This could be perfectly understood on the EuroParl task, as two language-dependent linguistic tools were employed for labelling. However, the FreeLing toolkit was used on EuTrans task for both languages, thus resulting quite disappointing that labels are not consistent inter languages.

5 Conclusions

This paper has presented a category-based approach to statistical machine translation, which is based on linguistic information. An integrated architecture, combining finite state transducers and stochastic dictionaries has been proposed. Some preliminary results are rather limited but also encouraging enough.

Acknowledgements

We would like to thank the people of the Spoken Language Systems Laboratory from the Instituto de Engenharia de Sistemas e Computadores I+D in Lisbon, for providing the linguistic analysis of the Portuguese Europarl.

References

A. de Gispert and J. B. Mariño 2006. *Linguistic knowledge in statistical phrase-based word alignment.* Natural Language Engineering (Vol 12, Issue 01, Pgs 91-108).

F. Casacuberta, E. Vidal and D. Picó. 2005. *Inference of finite-state transducers from regular languages.* Pattern Recognition (Vol 38, Num 9, Pgs 1431–1443).

J. González, A. L. Lagarda, J. R. Navarro, L. Eliodoro, A. Giménez, F. Casacuberta, J. M de Val, and F. Fabregat. 2006. *SisHiTra: a Spanish-to-Catalan hybrid machine translation system.* 5th SALTMIL Workshop on Minority Languages, Pgs 69-73, Genoa.

Koehn, P., Federico, M., Shen, W., Bertoldi, N., Hoang, H., Callison-Burch, C., Cowan, B., Zens, R., Dyer, C., Bojar, O., Moran, C., Constantin, A., and Herbst, E. 2006. *Open source toolkit for statistical machine translation: Factored translation models and confusion network decoding.* Technical report, John Hopkins University Summer Workshop.

M. Mohri, F. Pereira and M. Riley. 2002. *Weighted Finite-State Transducers in Speech Recognition.* Computer Speech and Language (Vol. 16, Num. 1, Pgs 69–88).

P. F. Brown, J. Cocke, S. A. Della Pietra, V. J. Della Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roosin. 1990. *A statistical approach to machine translation.* Computational Linguistics (Vol 16, Num. 2, Pgs. 79–85).

R. Zens, F. J. Och and H. Ney. 2002. *Phrase-based statistical machine translation.* `http://citeseer.nj.nec.com/zens02phrasebased.html`

S. Kumar, Y. Deng and W. Byrne. 2006. *A weighted finite state transducer translation template model for statistical machine translation.* Natural Language Engineering (Vol 12, Num 1, Pgs 35–75).

S. Niessen and H. Ney 2004. *Statistical Machine Translation with Scarce Resources Using Morpho-syntactic Information.* Computational Linguistics (Vol 30, Num 2, Pgs. 181-204).

X. Carreras, I. Chao, L. Padró and M. Padró 2004. *FreeLing: An Open-Source Suite of Language Analyzers.* Proceedings of the 4th International Conference on Language Resources and Evaluation, Lisbon.