Using ‘smart’ bilingual projection to feature-tag a monolingual dictionary

Katharina Probst
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
kathrin@cs.cmu.edu

Abstract

We describe an approach to tagging a monolingual dictionary with linguistic features. In particular, we annotate the dictionary entries with parts of speech, number, and tense information. The algorithm uses a bilingual corpus as well as a statistical lexicon to find candidate training examples for specific feature values (e.g. plural). Then a similarity measure in the space defined by the training data serves to define a classifier for unseen data. We report evaluation results for a French dictionary, while the approach is general enough to be applied to any language pair.

In a further step, we show that the proposed framework can be used to assign linguistic roles to extracted morphemes, e.g. noun plural markers. While the morphemes can be extracted using any algorithm, we present a simple algorithm for doing so. The emphasis hereby is not on the algorithm itself, but on the power of the framework to assign roles, which are ultimately indispensable for tasks such as Machine Translation.

1 Introduction and motivation

The Machine Translation community has recently undergone a major shift of focus towards data-driven techniques. Among these techniques, example-based (e.g. (Brown, 1997)) and statistical (e.g. (Brown et al., 1990; Brown et al., 1993)) are best known and studied. They aim at extracting information from bilingual text and building translation systems automatically. This empirical approach overcomes the development bottleneck that traditional transfer- and interlingua-based approaches face. What used to take years of human development time can now be achieved in a fraction of the time with similar accuracy. However, in studying such empirical approaches and the output of the resulting systems, there have been calls for the re-incorporation of more linguistic intuition and/or knowledge. One notable example in this context is (Yamada and Knight, 2001; Yamada and Knight, 2002), who introduce syntactic knowledge into their statistical translation model. Our approach goes in a similar direction. The AVENUE system (Carbonell et al., 2002) infers syntactic transfer rules similar to the ones a human grammar writer would produce. The training data is bilingual text, and learning is facilitated by the usage of linguistic information (e.g. parses, feature information). We focus primarily on a resource-rich/resource-poor situations, i.e. on language pairs where for one of the languages resources such as a parser are available, but not for the other language. It is outside the scope of this paper to describe our rule learning approach. The interested reader should refer to (Carbonell et al., 2002; Probst, 2002).

From the brief description above it should become apparent that heavy reliance on feature-tagged dictionaries and/or parsers becomes a new bottleneck for Machine Translation development. Our work focuses on target languages for which there does exist a dictionary, but its entries may are initially not tagged with linguistic feature values, so that the dictionary is a mere word list (which is what Example-based Machine Translation and Statistical Machine Translation systems use most frequently). Having the feature values can become very useful in translation. For example, if the English sentence contains a plural noun, the system can ensure that this word is translated into a plural noun in the target language (if the learned rule requires this).

Despite the importance of the feature tags, we cannot afford to build such a rich dictionary by hand. Moreover, we cannot even rely on the availability of experts that can write morphological rules for a given language. Rather, we wish to develop an algorithm that is general enough that it can be applied to any language pair and does not require knowledge of the target language's morphology.

In this paper, we explore the following features: parts of speech (*pos*), number on nouns, adjectives, and verbs, and tense on verbs. Furthermore, the process is fully automatic, thus eliminating the need for human expertise in a given language. Our main idea is based on using a bilingual corpus between English and a target language. In our experiments, we report results for French as the target language. We annotate the English side of the corpus with *pos* tags, using the Brill tagger (Brill, 1995). We further utilize a statistical bilingual (English→French) dictionary in order to find candidates translations for particular English words. Our work falls in line with the bilingual analysis described in (Yarowsky and Ngai, 2001; Yarowsky et al., 2001). While we use a different approach and tackle a different problem, the major reasoning steps are the same. (Yarowsky and Ngai, 2001) aim at *pos* tagging a target language corpus using English *pos* tags as well as estimation of lexical priors (i.e. what *pos* tags can a word have with what probability) and a tag sequence model. The authors further report results on matching inflected verb forms in the target language with infinitive verbs, as well as on noun phrase chunking. In all three cases, the information on the English side is used to infer linguistic information on the target language side. Our work follows the same idea.

2 Tagging the target language dictionary with *pos*

In a first step, we tag the target language dictionary entries with likely *pos* information. It is important to note that this is the first step in the process. The following steps, aiming at tagging entries with features such as number, are based on the possible *pos* assigned to the French entries.

We would like to emphasize clearly that the goal of our work is not 'traditional' *pos* tagging. Rather, we would like to have the target language dictionary tagged with likely *pos* tags, possibly more than one per word[1].

Having said this, we follow in principle the algorithm proposed by (Yarowsky and Ngai, 2001) to estimate lexical priors. We first find the most likely corresponding French word for each English word. Then we project the English *pos* onto the French word. While it is clear that words do not always translate into words of the same *pos*, the basic idea is that overall they are likely to transfer into the same *pos most of the time*. Using a large corpus will then give us averaged information on how often a word is the most likely correspondent of a noun, a verb, etc.

In this section, we restrict our attention (again, following (Yarowsky and Ngai, 2001)) to five 'core' *pos*, N (noun), V (verb), J (adjective), R (adverb), and I (preposition or subordinating conjunction). The algorithm was further only evaluated on N, V, and J, first because they are the most likely *pos* (so more reliable estimates can be given), and second because the remainder of the paper only deals with these three *pos*.

In preparation, we use the Brill tagger (Brill, 1995) to annotate the English part of the corpus with *pos* tags. Suppose we have an aligned bilingual sentence pair s_{e_j} – s_{f_j}. The algorithm then proceeds as follows: for each English word e_i in sentence s_{e_j} tagged with one of the core tags, look up all words in the statistical English→French dictionary that are proposedly translations of it. Then pick as a likely translation the word f_i with the highest probability in the statistical dictionary that *also* occurs in the French sentence s_{f_j}. We then simply add the number of times that a given word corresponds to an English word of tag t_k, denoted by $c(t_k, f_i)$. This information is used to infer $p(t_k|f_i)$:

$$p(t_k|f_i) = \frac{c(t_k, f_i)}{\sum_{j=1}^{m} c(t_j, f_i)}$$

Given the way our algorithm estimates the probabilities of *pos* for each French word, it is clear that some noise is introduced. Therefore, *any pos* will be assigned a non-zero probability by the algorithm for each French word. However, as was noted by (Yarowsky and Ngai, 2001), most words tend to have at most two *pos*. In an attempt to balance out the noise introduced by the algorithm itself, we do not want to assign more than two possible *pos* to each word. Thus, for each French word we only retain the two most likely tags and rescale their probabilities so that they sum to 1. Denote t_1 as the most likely tag for word f_i, t_2 as the second most likely tag for word f_i. Then

$$p(t_1|f_i)_{rescaled} = \frac{p(t_1|f_i)}{p(t_1|f_i) + p(t_2|f_i)}$$

and

$$p(t_2|f_i)_{rescaled} = \frac{p(t_2|f_i)}{p(t_1|f_i) + p(t_2|f_i)}.$$

In order to have a target language dictionary tagged with *pos*, we use the statistical bilingual dictionary and extract all French words. If a French word was encountered during the *pos* training, it is assigned the one or two most likely tags (together with the probabilities). Otherwise, the word remains untagged, but is retained in the target language dictionary.

In a second round of experiments, we slightly altered our algorithm. Instead of only extracting the most likely

[1] Each of the *pos* assignments is also annotated with a probability. The probabilities are not actually used in the work described here, but they can be utilized in the rule learning system.

	Nouns	Verbs	Adjectives
No Probabilities	79.448	81.265	70.707
With Probabilities	78.272	80.279	71.809

Table 1: Accuracies of *pos* Estimation for nouns, verbs, and adjectives, evaluated on 2500 French dictionary entries.

French correspondent, we take into account the correspondence probability as assigned by the statistical bilingual dictionary. Instead of simply counting how many times a French word corresponds to, say, a noun, the counts are weighted by the probability of each of the correspondences. The remainder of the algorithm proceeds as before.

We tested the algorithm on parts of the Hansard data, 200000 sentence pairs between English and French. The evaluation was done on 2500 French lexicon entries, which were hand-tagged with *pos*. For each automatically assignment *pos* assignment, we check whether this assignment was also given in the hand-developed partial dictionary. Partial matches are allowed, so that if the algorithm assigned one correct and one incorrect *pos*, the correctly assigned label is taken into account in the accuracy measure. Table 1 shows the results of the tagging estimates. It can be seen that due to the relative rarity of adjectives, the estimates are less reliable than for nouns and verbs. Further, the results show that incorporating the probabilities from the bilingual lexicon does not result in consistent estimation improvement. A possible explanation is that most of the French words that are picked as likely translations are highly ranked and correspond to the given English word with similar probabilities.

(Yarowsky and Ngai, 2001) propose the same algorithm as the one proposed here for their estimation of lexical priors, with the exception that they use automatic word alignments rather than our extraction algorithm for finding corresponding words. As for (Yarowsky and Ngai, 2001) estimating lexical priors is merely an intermediate step, they do not report evaluation results for this step. Further experiments should show what impact the usage of automatic alignments has on the performance of the estimation algorithm.

3 A feature value classifier

In this section, we describe the general algorithm of training a classifier that assigns a feature value to each word of a specific core *pos*. The following sections will detail how this algorithm is applied to different *pos* and different features. The algorithm is general enough so that it can be applied to various *pos*/feature combinations. The extraction of training examples is the only part of the process that changes when applying the algorithm to a different *pos* and/or feature.

3.1 Extraction of training data

Although the following sections will describe in greater detail how training data is obtained for each *pos*/feature combination, the basic approach is outlined here. As in the previous section, we use the sentence-aligned bilingual corpus in conjunction with the statistical bilingual dictionary to extract words that are likely to exhibit a feature. In the previous section, this feature was a particular *pos* tag. Here, we focus on other features, such as plural. For instance, when looking for plural nouns, we extract plural nouns from the English sentences (they are tagged as such by the Brill tagger, using the tag 'NNS'). We then extract the French word in the corresponding sentence that has the highest correspondence probability with the English word according to the statistical bilingual dictionary. This process again ensures that *most* (or at least a significant portion) of the extracted French words exhibits the feature in question. In principle, the purpose of the classifier training is then to determine what all (or most) of the extracted words have in common and what sets them apart.

3.1.1 Tagging of tense on verbs

The first feature we wish to add to the target language lexicon is tense on verbs. More specifically, we restrict our attention to PAST vs. NON-PAST. This is a pragmatic decision: the tagged lexicon is to be used in the context of Machine Translation, and the most common two tenses that Machine Translation systems encounter are past and present. In the future, we may investigate a richer tense set.

In order to tag tense on verbs, we proceed in principle as was described before when estimating lexical priors. We consider each word e_i in the English corpus that is tagged as a past tense verb. Then we find the likely correspondence on the French side, f_i, by considering the list of French words that correspond to the given English word, starting from the pair with the highest correspondence probability (as obtained from the bilingual lexicon). The first French word from the top of the list that also occurs in the French sentence is extracted and added to the training set:

$$f_i = argmax_{f_j, 0<j<m} p(f_j|e_i),$$

where m is the number of French words in the lexicon.

3.1.2 Tagging of number on nouns, adjectives, and verbs

Further, we tag nouns with number information. Again, we restrict our attention to two possible values: SINGULAR vs. PLURAL. Not only does this make sense from a pragmatic standpoint (i.e. if the Machine Translation system can correctly determine whether a word should be singular or plural, much is gained); it also allows us to train a binary classifier, thus simplifying the problem.

The extraction of candidate French plural nouns is done as expected: we find the likely French correspondent of each English plural noun (as specified by the English *pos*-tagger), and add the French words to the training set.

However, when tagging number on adjectives and verbs, things are less straight-forward, as these features are not marked in English and thus the information cannot be obtained from the English *pos*-tagger. In the case of verbs, we look for the first noun *left* of the candidate verb. More specificially, we consider an English verb from the corpus only if the closest noun to the left is tagged for plural. This makes intuitive sense linguistically, as in many cases the verb will follow the subject of a sentence.

For adjectives, we apply a similar strategy. As most adjectives (in English) appear directly before the noun that they modify, we consider an adjective only if the closest noun to the *right* is in the plural. If this is the case, we extract the likely French correspondent of the adjective as before.

3.2 Projection into a similarity space of characters

The extracted words are then re-represented in a space that is similar in concept to a vector space. This process is done as follows: Let

$$F_{(\langle pos\rangle,\langle feature\rangle)} = \{f_1, f_2, ..., fn\}$$

denote the set of French words that have been extracted as training data for a particular *pos*/feature combination. For notational convenience, we will usually refer to $F_{(\langle pos\rangle,\langle feature\rangle)}$ as F in the remainder of the paper. The reader is however reminded that each F is associated with a particular *pos*/feature combination. Let

$$max\_length = max_{i:0<i<n}|f_i|$$

denote the length of the longest word in $F_{(\langle pos\rangle,\langle feature\rangle)}$. Then we project all words in this set into a space of max_length dimensions, where each character index represents a dimension. This implies that for the longest word (or all words of length max_length), each character is one dimension. For shorter words, the projection will contain empty dimensions. Our idea is based on the fact that in many languages, the most common morphemes are either prefixes or suffixes. We are interested in comparing what most words in F begin or end in, rather than emphasizing on the root part, which tends to occur inside the word. Therefore, we simply assign an empty value ('-') to those dimensions for short words that are in the middle of the word. A word f_i, such that $|f_i| < max\_length$, is split in the middle and its characters are assigned to the dimensions of the current space from both ends. In case $|f_i| = 2k+1, k \in \mathbb{R}^+$, we double the character at position $\lceil |f_i|/2 \rceil$, so that it can potentially be part of a suffix or a prefix.

For example if $F_{(\langle pos\rangle,\langle feature\rangle)} = \{droits, ils, femmes, ..., orateurs\}$, then the corresponding space will be represented as follows:

d	r	o	-	-	i	t	s
i	l	-	-	-	-	l	s
f	e	m	-	-	m	e	s
...							
o	r	a	t	e	u	r	s

3.3 Similarity measure

In order to determine what the common feature between most of the words in $F_{(\langle pos\rangle,\langle feature\rangle)}$ is, we define a similarity measure between any two words as represented in the space.

We want our similarity measure to have certain properties. For instance, we want to 'reward' (consider as increasing similarity) if two words have the same character in a dimension. By the same token, a different character should decrease similarity. Further, the empty character should not have any effect, even if both words have the empty character in the same dimension. Regarding the empty character a match would simply consider short words similar, clearly not a desired effect.

We therefore define our similarity measure as a measure related to the inner product of two vectors $\langle x, y\rangle = \sum_{h=1}^{k} x_h y_h$, where k is the number of dimensions. Note however two differences: first, the product $x_h y_h$ is dependent on the specific vector pair. It is defined as

$$x_h y_h = \begin{cases} 1, & x_h = y_h, x_h \neq \text{'-'} \\ 0, & \text{otherwise} \end{cases}$$

Second, we must normalize the measure by the number of dimensions. This will become important later in the process, when certain dimensions are ignored and we do not always compute the similarity over the same number of dimensions. The similarity measure then looks as follows:

$$sim(x, y) = \frac{\sum_{h=1}^{k} x_h y_h}{k},$$

Note that when all dimensions are considered, k will correspond to max_length. The similarity measure is computed for each pair of words $f_i, f_j \in F_{(\langle pos\rangle,\langle feature\rangle)}, i \neq j$. Then the average is computed.

This number can be regarded as a measure of the incoherence of the space:

$$incoh_F = \frac{\sum_{i,j \in F: i \neq j} sim(f_i f_j)}{\frac{1}{2} * \binom{n}{k}}$$

Although it seems counterintuitive to define an *in*coherence measure as opposed to a coherence measure, calling the measure an incoherence measure fits with the intuition that low incoherence corresponds to a coherent space.

4 Run-time classification

4.1 Perturbing and unifying dimensions

The next step in the algorithm is to determine what influence the various dimensions have on the coherence of the space. For each dimension, we determine its impact: does it increase or decrease the coherence of the space. To this end, we compute the incoherence of the space with one dimension blocked out at a time. We denote this new incoherence measure as before, but with an additional subscript to indicate which dimension was blocked out, i.e. disregarded in the computation of the incoherence. Thus, for each dimension $i, 1 < i < max\_length$, we obtain a new measure $incoh_{F,i}$. Two things should be noted: first, $incoh_{F,i}$ measures the coherence of the space *without* dimension i. Further, the normalization of the similarity metric becomes important now, if we want to be able to compare the incoherence measures.

In essence, the impact of a dimension is perturbing if disregarding it increases the incoherence of the space. Similarly, it is unifying if its deletion decreases the incoherence of the space. The impact of a dimension is measured as follows:

$$imp_{F,i} = \frac{(incoh_{F,i} - incoh_F)}{incoh_F}$$

We then conjecture that those dimensions whose impact is positive (i.e. disregarding it results in an increased incoherence score) are somehow involved in marking the feature in question. These features, together with their impact score $imp_{F,i}$ are retained in a set

$$OptSet_F = \{i | 1 < i < max\_length, imp_{F,i} > 0\}.$$

The $OptSet_F$ is used for classification as described in the following section.

4.2 Classification of French dictionary entries

From the start we have aimed at tagging the target language dictionary with feature values. Therefore, it is clearly not enough to determine which dimensions in the space carry information about a given feature. Rather, we use this information to classify words from the target language dictionary.

To this end, all those words in the target language dictionary that are of the *pos* in question are classified using the extracted information (the reader is reminded that the system learns a classifier for a particular *pos*/feature combination). For a given word f_{test}, we first project the word into the space defined by the training set. Note that in can happen that $|f_{test}| > max\_length$, i.e. that f_{test} is longer than any word encountered during training. In this case, we delete enough characters from the *middle* of the word to fit it into the space defined by the training set. Again, this is guided by the intuition that often morphemes are marked at the beginning and/or the end of words. While the deletion of characters (and thus elimination of information) is theoretically a suboptimal procedure, it has a negligible effect at run-time.

After we project f_{test} into the space, we compute the coherence of the combined space defined by the set denoted by $\{F, f_{test}\} = F \bigcup f_{test}$ as follows, where the similarity is computed as above and n again denotes the size of the set F:

$$incoh_{\{F,f_{test}\}} = \frac{\sum_{i \in F} sim(f_{test}, f_i)}{n}$$

In words, the test word f_{test} is compared to each word f_i in the set F.

In the following, all dimensions $i \in OptSet_F$ are blocked out in turn, and $incoh_{\{F,f_{test}\},i}$ is computed, i.e. the incoherence of the set $\{F, f_{test}\}$ with one of the dimensions blocked out. As before, the impact of dimension is defined by

$$imp_{\{F,f_{test}\},i} = \frac{(incoh_{\{F,f_{test}\},i} - incoh_{\{F,f_{test}\}})}{incoh_{\{F,f_{test}\}}}$$

Finally, the word f_{test} is classified as *'true'* (i.e. as exhibiting the feature) if blocking out the dimensions in $OptSet_F$ descreases incoherence more than the average, i.e. when the incoherence measures were computed on the training set. Thus, the final decision rule is:

$$f_{test} = \begin{cases} true, & \sum_{i \in OptSet_{\{F,f_{test}\}}} imp_{\{F,f_{test}\},i} \\ & > \sum_{i \in OptSet_F} imp_{F,i} \\ false, & \text{otherwise} \end{cases}$$

In practice, this decision rule has the following impact: If, for instance, we wish to tag nouns with plural information, a word f_{test} will be tagged with plural if classified as true, with singular if classified as false.

5 Experimental results

As with *pos* estimation, we tested the feature tagging algorithms on parts of the Hansards, namely on 200000

	No Probs	With Probs
N: Pl vs. Sg	95.584	95.268
J: Pl vs. Sg	97.143	97.037
V: Pl vs. Sg	85.075	85.019
V: Past vs. Non-Past	72.832	73.043

Table 2: Accuracies of tagging nouns, adjectives, and verbs with plural or singular, and tagging verbs with past vs. non-past, based on two dictionaries that was tagged with *pos* automatically, one of which used the probabilities of the translation dictionary for *pos* estimation.

sentence pairs English-French. Accuracies were obtained from 2500 French dictionary entries that were not only hand-tagged with *pos*, but also with tense and number as appropriate. Table 2 summarizes the results. As mentioned above, we tag nouns, adjectives, and verbs with PLURAL vs. SINGULAR values, and additionally verbs with PAST vs. NON-PAST information. In order to abstract away from *pos* tagging errors, the algorithm is only evaluated on those words that were assigned the appropriate *pos* for a given word. In other words, if the test set contains a singular noun, it is looked up in the automatically produced target language dictionary. If this dictionary contains the word as an entry tagged as a noun, the number assignment to this noun is checked. If the classification algorithm assigned singular as the number feature, the algorithm classified the word successfully, otherwise not. This way, we can disregard *pos* tagging errors.

When estimating *pos* tags, we produced two separate target language dictionaries, one where the correspondence probabilities in the bilingual English→French dictionary were ignored, and one where they were used to weigh the correspondences. Here, we report results for both of those dictionaries. Note that the only impact of the a different dictionary (automatically tagged with *pos* tags) is that the test set is slightly different, given our evaluation method as described in the previous paragraph. The fact that evaluating on a different dictionary has no consistent impact on the results only shows that the algorithm is robust on different test sets.

The overall results are encouraging. It can be seen that the algorithm very successfully tags nouns and adjectives for plural versus singular. In contrast, tagging verbs is somewhat less reliable. This can be explained by the fact that French tags number in verbs differently in different tenses. In other words, the algorithm is faced with more inflectional paradigms, which are harder to learn because the data is fragmented into different patterns of plural markings.

A similar argument explains the lower results for past versus non-past marking. French has several forms of past, each with different inflectional paradigms. Further, different groups of verbs inflect for tense differntly, fragmenting the data further.

6 Morpheme role assignment

While in this work we use the defined space merely for classification, our approach can also be used for assigning roles to morphemes. Various morpheme *extraction* algorithms can be applied to the data. However, the main advantage of our framework is that it presents the morphology algorithm of choice with a training set for particular linguistic features. This means that whatever morphemes are extracted, they can *immediately* be assigned their linguistic roles, such as number or tense. Role assignment is generally not focused on or excluded entirely in morphology learning. While mere morpheme extraction is useful and sufficient for certain tasks (such as root finding and stemming), for Machine Translation and other tasks involving deeper syntactic analysis it is not enough to find the morphemes, unless they are also assigned roles. If, for instance, we are to translate a word for which there is no entry in the bilingual dictionary, but by stripping off the plural morpheme, we can find a corresponding (singular) word in the other language, we can ensure that the target language word is turned into the plural by adding the appropriate plural morpheme.

In this section, we present *one* possible algorithm for extracting morphemes in our framework. Other, more sophisticated, unsupervised morphology algorithms, such as (Goldsmith, 1995), are available and can be applied here. Staying within our framework ensures the additional benefit of immediate role assignment.

Another strength of our approach is that we make no assumption about the contiguity of the morphemes. Related work on morphology generally makes this assumption (e.g. (Goldsmith, 1995)), with the notable exception of (Schone and Jurafsky, 2001). While in the current experiments the non-contiguity possibility is not reflected in the results, it can become important when applying the algorithm to other languages such as German.

We begin by conjecturing that most morphemes will not be longer than four characters, and learn only patterns up to that length. Our algorithm starts by extracting all patterns in the training data of up to four characters, however restricting its attention to the dimensions in $OptSet_F$. If $OptSet_F$ contains more than 4 dimensions, the algorithm works only with those 4 dimensions that had the greatest impact. All 1, 2, 3, and 4 character combinations that occur in the training data are listed together with how often they occur. The reasoning behind this is that those patterns that occur most frequently in the training data are likely those ‘responsible’ for marking the given feature.

However, it is not straightforward to determine automatically how long a morpheme is. For instance, consider the English morpheme '-ing' (the gerund morpheme). The algorithm will extract the patterns 'i__', '_n_', '__g', 'in_', 'i_g', '_ng', and 'ing'. If we based the morpheme extraction merely on the frequency of the patterns, the algorithm would surely extract one of the single letter patterns, since they are guaranteed to occur at least as many times as the longer patterns. More likely, they will occur more frequently. In order to overcome this difficulty, we apply a subsumption filter. If a shorter patterns is subsumed by a longer one, we no longer consider it. Subsumption is defined as follows: suppose pattern α_i appears with frequency c_{α_i}, where as pattern α_j appears with frequency c_{α_j}, and that α_i is shorter than α_j. Then α_i is subsumed by α_j if

$$\frac{c_{\alpha_j}}{c_{\alpha_i}} > \frac{1}{2}.$$

The algorithm repeatedly checks for subsumption until no more subsumptions are found, at which point the remaining patterns are sorted by frequency. It then outputs the most frequent patterns. The cutoff value (i.e. how far down the list to go) is a tunable parameter. In our experiments, we set this parameter to 0.05 probability. Note that we convert the frequencies to probabilities by dividing the counts by the sum of all patterns' frequencies.

The patterns are listed simply as arrays of 4-characters (or fewer if $OptSet_F$ contains fewer elements). It should be noted that the characters are listed in the order of the dimensions. This, however, does not mean that the patterns have to be contiguous. For instance, if dimension 1 has a unifying effect, and so do dimensions 14, 15, and 16, the patterns are listed as 4-character combinations in increasing order of the dimensions.

For illustration purposes, table 3 lists several patterns that were extracted for past tense marking on verbs[2]. All highly-ranked extracted patterns contained only letters in the last two dimensions, so that only those two dimensions are shown in the table.

Further investigation and the development of a more sophisticated algorithm for extracting patterns should enable us to collapse some of the patterns into one. For instance, the patterns 'ée' and 'és' should be considered special cases of 'é_'. Note further that the algorithm extracted the pattern '_s', which was caused by the fact that many verbs were marked for plural in the passé composé in French. In order to overcome this difficulty, a more complex morphology algorithm should combine findings from different *pos*/feature combinations. This has been left for future investigation.

dimension $k-1$	dimension k
é	e
é	s
é	_
_	é
_	s

Table 3: Sample morpheme patterns extracted for past tense markers on verbs. For this run, $k = 15$. Only the last two dimensions are shown. No extracted pattern involved any of the other dimensions.

7 Discussion and conclusion

We have presented an approach to tagging a monolingual dictionary with linguistic features such as *pos*, number, and tense. We use a bilingual corpus and the English *pos* tags to extract information that can be used to infer the feature values for the target language.

We have further argued that our approach can be used to infer the morphemes that mark the linguistic features in question *and* to assign the morphemes linguistic meaning. While various frameworks for unsupervised morpheme extraction have been proposed, many of them more sophisticated than ours, the main advantage of this approach is that the annotation of morphemes with their meaning is immediate. We believe that this is an important contribution, as role assignment becomes indispensible for tasks such as Machine Translation.

One area of future investigation is the improvement of the classification algorithm. We have only presented one approach to classification. In order to apply established algorithms such as Support Vector Machines, we will have to adopt our algorithm to extract a set of likely positive examples as well as a set of likely negative examples. This will be the next step in our process, so that we can determine the performance of our system when using various well-studied classification methods.

This paper represents our first steps in bilingual feature annotation. In the future, we will investigate tagging target language words with gender and case. This information is not available in English, so it will be a more challenging problem. The extracted training data will have to be fragmented based on what has already been learned about other features.

We believe that our approach can be useful for any application that can gain from linguistic information in the form of feature tags. For instance, our system (Carbonell et al., 2002) infers syntactic transfer rules, but it relies heavily on the existence of a fully-inflected, tagged target language dictionary. With the help of the work described here we can obtain such a dictionary for any language for which we have a bilingual, sentence-aligned corpus. Other approaches to Machine Translation as well as ap-

[2] Note that no morphemes for the imparfait were extracted. This is an artifact of the training data which contains very few instances of imparfait.

plications like shallow parsing could also benefit from this work.

References

Eric Brill. 1995. *Transformations-based error-driven learning and natural language processing: A case study in part of speech tagging.* Computational Linguistics, 16(2):29-85.

Peter Brown, J. Cocke, V.D. Pietra, S.D. Pietra, J. Jelinek, J. Lafferty, R. Mercer, and P. Roossina. 1990. *A statistical approach to Machine Translation.* Computational Linguistics, 16(2):79-85.

Peter Brown, S.D. Pietra, V.D. Pietra, and R. Mercer. 1993. *The mathematics of statistical Machine Translation: Parameter estimation.* Computational Linguistics,19(2):263-311.

Ralf Brown. 1997. *Automated Dictionary Extraction for 'Knowledge-Free' Example-Based Translation*. Proceedings of the 7th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-97), pp. 111-118.

Jaime Carbonell, Katharina Probst, Erik Peterson, Christian Monson, Alon Lavie, Ralf Brown, and Lori Levin. 2002. *Automatic Rule Learning for Resource-Limited MT.* Proceedings of the 5th Biennial Conference of the Association for Machine Translation in the Americas (AMTA-02).

John Goldsmith. 1995. *Unsupervised Learning of the Morphology of a Natural Language.* Computational Linguistics 27(2): 153-198.

Katharina Probst. *Semi-Automatic Learning of Transfer Rules for Machine Translation of Low-Density Languages.* Proceedings of the Student Session at the 14th European Summer School in Logic, Language and Information (ESSLLI-02).

Patrick Schone and Daniel Jurafsky. 2001. *Knowledge-Free Induction of Inflectional Morphologies.* Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL-01).

Kenji Yamada and Kevin Knight. 2002. *A Decoder for Syntax-Based Statistical MT.* Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL-02).

Kenji Yamada and Kevin Knight. 2001. *A Syntax-Based Statistical Translation Model.* Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL), 2001.

David Yarowsky and Grace Ngai. 2001. *Inducing Multilingual POS Taggers and NP Bracketers via Robust Projection Across Aligned Corpora.* In Proceedings of the Second Meeting of the North American Chapter or the Association for Computational Linguistics (NAACL-2001), pp. 200-207.

David Yarowsky, Grace Ngai, and Richard Wicentowski. 2001. *Inducing Multilingual Text Analysis Tools via Robust Projection across Aligned Corpora.* Proceedings of the First International Conference on Human Language Technology Research (HLT-01).