Tagset Mapping and Statistical Training Data Cleaning-up

Felix Pîrvan, Dan Tufiş
Research Institute for Artificial Intelligence, Romanian Academy
13, Calea 13 Septembrie, 050711, Bucharest
barbacot@yahoo.com, tufis@racai.ro

Abstract

The paper describes a general method (as well as its implementation and evaluation) for deriving mapping systems for different tagsets available in existing training corpora (gold standards) for a specific language. For each pair of corpora (tagged with different tagsets), one such mapping system is derived. This mapping system is then used to improve the tagging of each of the two corpora with the tagset of the other (this process will be called cross-tagging). By reapplying the algorithm to the newly obtained corpora, the accuracy of the underlying training corpora can also be improved. Furthermore, comparing the results with the gold standards makes it possible to assess the distributional adequacy of various tagsets used in processing the language in case. Unlike other methods, such as those reported in (Brants, 1995) or (Tufiş & Dragomirescu, 2004), which assume a subsumption relation between the considered tagsets, and as such they aim at minimizing the tagsets by eliminating the feature-value redundancy, this method is applicable for completely unrelated tagsets. Although the experiments were focused on morpho-syntactic (POS) tagging, the method is applicable to other types of tagging as well.

1. Introduction

In processing parallel corpora, it is desirable to use the same encoding system for all the languages (e.g. Multext-East, http://nl.ijs.si/ME/V3/msd/html/msd.html, which is underlying all our multilingual resources). However, more often than not, the freely available training data are using different and unrelated tagsets.

Given two different reference corpora (for the same language), each tagged with its own tagset and called from now on the gold standard for that tagset, one could ask several legitimate questions:

a. Is it possible to merge the two corpora and have the larger corpus confidently tagged with either of the tagsets?
b. Can the objective stated above be satisfactorily achieved by simply building a language model from one corpus and then use it to tag the other (using whatever tagger)?
c. Can the two gold standards be improved?

We will describe a method showing that the answer to the first question is positive. Moreover, repeating the procedure for other pairs of corpora and tagsets would allow us to build a much larger corpus, tagged in parallel with all the tagsets of its initial components, and use it as a gold standard for each of those tagsets.

Concerning the second question, the positive answer depends on the dimensions of the two corpora used for building the language models; yet we argue that the method we will describe in this paper yields better results. Why? It is simply more informed, making use of the fact that the corpus to tag with a new tagset is already tagged with its own tagset.

The answer to the third question is again positive: the proposed method allows for re-tagging any of the gold standards with their original tagsets and, by comparing the two versions of the corpus, the systematic errors can be easily spotted and removed.

For the experiments reported herein, we used the "1984" MULTEXT-EAST reference multilingual corpus (the English component of it) and a comparable-size subset of the SemCor corpus (see Chapter 5).

2. Overview

We will call *direct tagging* (DT) the usual process of tagging, where a language model learnt from a gold standard corpus is used in POS-tagging of a different corpus. Tagging the same corpus used for language model learning is called *biased tagging* (BT). With a consistently tagged gold standard, the biased tagging is expected to be almost identical to the one in the gold standard. We will use this observation to evaluate the gold standards improvements after applying our method.

The *cross-tagging* (CT) is a method that, given two corpora, each tagged with different tagsets, produces the two corpora tagged with the other one's tagset, using a mapping system between the two tagsets. The cross-tagging is a stochastic process, it uses language models learnt from the corpora involved.

We denote by $A_{GS}(X)$ the *A gold standard corpus* which is tagged in terms of the *X tagset* and by $B_{GS}(Y)$ the *B gold standard corpus* which is tagged in terms of the *Y tagset*.

With these notations, the processes of direct, biased and cross-tagging can be represented as below:

DT: $A_{GS}(X) + B \rightarrow B_{DT}(X)$
$B_{GS}(Y) + A \rightarrow A_{DT}(Y)$
BT: $A_{GS}(X) + A \rightarrow A_{BT}(X)$
$B_{GS}(Y) + B \rightarrow B_{BT}(Y)$
CT: $A_{GS}(X)+A_{DT}(Y)+B_{GS}(Y)+B_{DT}(X) \rightarrow A_{CT}(Y)+B_{CT}(X)$

We claim that the cross-tagged versions $A_{CT}(Y)$, $B_{CT}(X)$ will be more accurate than the direct-tagged ones $A_{DT}(Y)$ and $B_{DT}(X)$ respectively.

The cross-tagging works with both the gold standard and the direct-tagged versions of the two corpora and involves two main steps: building a mapping system between the two tagsets and improving the direct tagged versions using this mapping system. To obtain the direct-tagged corpora, any tagger can be used. In our experiments, we used the TnT tagger developed by T. Brants (2000).The overall system architecture is shown in Figure 1. The chapters 3 and 4 describe in details the major steps of the cross-tagging process.

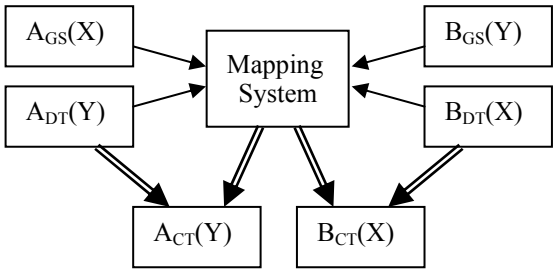

Figure 1. System Architecture

3. Tagset Mapping

From two versions of the same corpus, each tagged with a different tagset, we will extract a map between the two tagsets. Having two corpora, we can obtain two such *corpus-specific maps*, which will then be merged into a single, and of higher confidence, global map. This process can be represented as below.

$$A_{GS}(X)+A_{DT}(Y)\rightarrow M_A(X,Y)\&B_{GS}(Y)+B_{DT}(X)\rightarrow M_B(X,Y)$$
$$M_A(X, Y) + M_B(X, Y) \rightarrow M(X, Y)$$

In the next two sections, we present the way we obtained the global map.

3.1. The Partial Maps

Let $X = \{x_1, x_2, \ldots, x_n\}$ and $Y = \{y_1, y_2, \ldots, y_m\}$ be the two tagsets. For a corpus tagged with both X and Y tagsets, we can build the *contingency table* shown in Table 1.

	y_1	y_2	…	y_m	
x_1	N_{11}	N_{12}	…	N_{1m}	N_{x1}
x_2	N_{21}	N_{22}	…	N_{2m}	N_{x2}
…	…	…	…	…	…
x_n	N_{n1}	N_{n2}	…	N_{nm}	N_{xn}
	N_{y1}	N_{y2}	…	N_{ym}	N

Table 1. The <X,Y> Contingency Table

The symbols have the following meanings:
N_{ij} – the number of tokens tagged both with x_i and y_j;
N_{xi} – the number of tokens tagged with x_i;
N_{yj} – the number of tokens tagged with y_j;
N – the total number of tokens in the corpus.

For each tag $x \in X$, we define a subset of Y, let it be $Y_x \subseteq Y$, that has the property that for any $y_j \in Y_x$ and for any $y_k \in Y - Y_x$, the probability of x conditioned by y_j is significantly higher than the probability of x conditioned by y_k. We say that x *is preferred* by tags in Y_x, or conversely, that tags in Y_x *prefer* x.

Let $PSet(x_i)$ be the set of probabilities of $x_i \in X$, conditioned by each $y \in Y$. It can be expressed as below.

$$PSet(x_i) = \{p(x_i|y_j) \mid y_j \in Y\},$$
$$\text{where } p(x_i|y_j) = p(x_i,y_j) / p(y_j) \cong N_{ij} / N_{yj}$$

Now, to find the values in $PSet(x_i)$ significantly higher than the others means to divide $PSet(x_i)$ in two clusters. The most significant cluster (MSC), i.e. the cluster containing the greater values, will give us Y_x, as shown below.

$$Y_x = \{y \in Y \mid p(x|y) \in MSC(P(x))\}$$

The clustering algorithm chosen (but any other may be used) is of single-link type, based on the raw distance between the values. This type of algorithm offers a fast top-down approach (remember that we only need two final clusters): sort the values in descending order, find the greatest distance between two consecutive values and split the values at that point. If more than one such greatest distance exists, the one between the smaller values is chosen to split on. The elements N_{ij} of the contingency table define a sparse matrix, with most of the values to cluster being zero. However, at least one value will be non-zero. Thus the most significant cluster will never contain zeroes, but it may contain all the non-zero values.

Let's look at an example. From the fragment of contingency table in Table 2, we can deduce the following:

$$PSet(x_1)=\{0.8, 0.05, 1\};\ MSC(P(x_1))=\{0.8, 1\}$$
$$\Rightarrow Y_{x1}=\{y_1, y_3\}$$

	y_1	y_2	y_3	
x_1	80	50	5	135
…	…	…	…	…
	100	1000	5	1105

Table 2. A Contingency Table Example

The preference relation is a first level filtering of the tag mappings for which insufficient evidence is provided by the gold standard corpora. This filtering would eliminate several real wrong mappings (not all of them) but also could leave out correct mappings that occurred much less frequently than others did. We will address this issue in the next section.

A *partial map* from X to Y (denoted PM_X) is defined as the set of tag pairs $(x,y) \in X \times Y$ for which y prefers x. Similarly a partial map from Y to X (denoted by PM_Y) can be defined. They can be expressed as below.

$$PM_X(X, Y) = \{(x, y) \in X \times Y \mid y \in Y_x\}$$
$$PM_Y(X, Y) = \{(x, y) \in X \times Y \mid x \in X_y\}$$

3.2. The Global Map

The two partial maps for each corpus may be merged into a single global map. We want to filter out all the false positives the partial maps might contain, while reducing the false negatives as much as possible. We tried several methods and we came up with the following merging formulae.

$$M_A(X, Y) = PM_{AX}(X, Y) \cup PM_{AY}(X, Y)$$
$$M_B(X, Y) = PM_{BX}(X, Y) \cup PM_{BY}(X, Y)$$
$$M(X, Y) = M_A(X, Y) \cap M_B(X, Y)$$

The maps computed in the first two formulae above will be referred as corpus-specific maps, while the other one is the targeted global map itself. The global map contains all the tag pairs for which one of the tags prefers the other, in both corpora. As this condition is a very strong one, several possibly correct mappings will be left out from M(X,Y) either because of insufficient data, or because of idiosyncratic behaviour of some lexical items. To correct this problem the global map is supplemented with the token maps.

3.3. The Token Maps

The global map expresses the preferences from one tag to another in a non-lexicalized way and it is used as a back-off mechanism when a more precise type of mapping is not possible, namely the lexicalized mapping. The

data structures for lexicalized mappings are called *token maps*. They are built only for the token types, common to both corpora (except for happax legomena).

The tokens types that occur only in one corpus will be mapped via the global map. The global map is also used for dealing with token types occurring in one corpus in contexts dissimilar to any context of occurrence in the other corpus.

3.3.1. Provisional Token Maps

For each common token type, we first build a *provisional token map* the same way we built the global map, that is, build contingency tables, extract partial maps from them, and then merge those partial maps.

Example: Building a Provisional Token Map. The token type *will* has the contingency tables shown in Table 3 and Table 4. The full-zero rows and columns (i.e. tags never assigned to *will*) were dropped for convenience.

will	MD	VB	NN	
VMOD	170	1	1	172
NN	2	1	4	7
	172	2	5	179

Table 3. Contingency table of *will* for the 1984 corpus

will	VMOD	NN	
MD	236	1	237
VB	28	0	28
NN	0	4	4
	264	5	269

Table 4. Contingency table of *will* for a fragment of the SemCor corpus

The tags have the following meanings:
VMOD, MD – modal verb
NN (both tagsets) – noun
VB – verb, base form

Each table has the rows marked with the tags from the gold standard version and the columns with the tags of the direct-tagged version.

The provisional token map extracted from these tables is:

$M_{will}(1984, SemCor) = \{(VMOD, MD), (NN, NN)\}$

It can be observed that the tag VB of the SemCor tagset is not mapped in this phase.

A consistent tagged corpus assumes that a word occurring in similar contexts should be identically tagged. We say that a tag marks the class of contexts in which a word was systematically labelled by it.

If a word *w* of a two-way tagged corpus is tagged by the pair <x,y> and this pair belongs to $M_w(X,Y)$ it means that the there are contexts marked by x similar with some contexts marked by y. If <x,y> does not belong to $M_w(X,Y)$ there are two possible situations:
a) either x or y (or both) are unmapped.
b) both x and y are mapped to some other tags

In the next sub-section we discuss the first case. The second one will be addressed in section 4.1.

3.3.2. Unmapped Tags

A tag unmapped for a specific token type may mean one of two things: either none of the contexts it marks is observed in the other corpus, or the tag is wrongly assigned for that particular token type.

The second possibility brings up one of the goals of this paper, that is, to improve the quality of the gold standards.

If we decide that the unmapped tag was wrongly assigned to the current token, the only thing to do is to trust the direct tagging and leave the tag unmapped.

In order to decide when it is likely to have a new context and when it is a wrong assignment, we relied on empirical observations that led us to the conclusion that the more frequent the token type appears in the other corpus, the less likely is for a tag, unmapped at token level, to mark a new context. To establish the threshold above which we considered a wrong assignment situation, we used the following empirical method. For each case of a tag unmapped at token level, we stored in an array the logarithm of the frequency of that token in the other corpus. The chosen threshold value is the one having the property that the sum of all smaller values in the array is closest to the sum of all greater values. The threshold frequency is the closest integer to the exponential of that value.

With the values in Table 3 for our example, the frequency of *will* in the 1984 corpus (179) is well above the threshold frequency (60). We therefore consider that all the contexts for *will* should have been observed and the unmapped tag VB does not mark a new context, but rather it represents a wrong assignment, so it is left unmapped.

The unmapped tags assigned to tokens with frequency below the threshold may signal the occurrence of the respective tokens in new contexts. If this is true, these tags will be mapped using the global map. To find out whether the new context hypothesis is acceptable, we use another heuristics based on the notion of tag sympathies.

3.3.3. Tag Sympathies

Given a tagged corpus, we define the *sympathy between two tags* x_1 and x_2, of the same tagset, written $S(x_1,x_2)$, as the number of token types having at least one occurrence tagged x_1 and at least one occurrence tagged x_2. By definition, the sympathy between a tag and itself is infinite. The relation of sympathy is symmetrical.

During the direct tagging, the tokens are usually tagged only with tags from the ambiguity classes learnt from the gold standard corpus. Therefore, if a specific token appears in context unseen during the LM construction, it will be inevitably wrongly tagged during the direct tagging. This error would show up because this tag, *x*, and the one in the gold standard, *y*, are very likely not to be mapped to each other in the map of the current token. If y is not mapped at all in the token's map, the algorithm checks if the tags mapped to y in the global map are sympathetic to any tag in the ambiguity class of the token type considered.

Some examples of highly sympathetic morphological categories for the English language are: nouns and base form verbs, past tense verbs and past participle verbs, adjectives and adverbs, nouns and adjectives, nouns and present participle verbs, adverbs and prepositions.

Example: Token Map Based on Tag Sympathies. The

token type *behind* has the contingency tables shown in Table 5 and Table 6.

behind	IN	
PREP	41	41
ADVE	9	9
	50	50

Table 5. Contingency table of *behind* for the 1984 corpus

behind	PREP	
IN	5	5
	5	5

Table 6. Contingency table of *behind* for a fragment of the SemCor corpus

The provisional token map is:

M_{behind}(1984, SemCor) = {(PREP, IN)}

There is one unmapped tag: ADVE. The global map M contains two mappings for ADVE:

M(ADVE)={RB, RBR}

The involved sympathies:

S(RB, IN) = 59, S(RBR, IN)=0

The sympathy relation being relevant only for the first pair, the token map for *behind* will become:

M_{behind}(1984, SemCor) = {(PREP, IN), (ADVE, RB)}

This new map will allow for the automatic corrections of the direct tagging of various occurrences of the token *behind.*

We described the construction of the mapping data structures, composed of one global map and many token maps. We now move on to the second step of the cross-tagging process, discussing how the mapping data structures are used.

4. Improving the Direct-Tagged Versions of the Two Corpora

To improve the direct-tagged version of a corpus, we go through two stages: first identifying the errors and then correct them. Obviously not all the errors can be automatically identified and not all the changes are correct, but the overall accuracy will nevertheless be improved. In the next section we describe how the plausible errors are spotted.

4.1. Error Identification

We have two direct-tagged corpora, $A_{DT}(Y)$ and $B_{DT}(X)$. They are treated independently, so we will further analyze only one of them, let it be $A_{DT}(Y)$. For each token of this corpus, we must decide if it was correctly tagged. Suppose the token w_k is tagged x in $A_{GS}(X)$ and y in $A_{DT}(Y)$. If the token type of that token, let it be w, has a token map, then we use the token map, otherwise, we use the global map. Let M_c be the chosen map.

If x is not mapped in M_c, or if $(x,y)\in M_c$, no action is taken. In the latter case, the direct tagging is in full agreement with the map, while in the former, the direct tagging is considered correct having no proof otherwise.

If the conditions above are not met, that is, if x is mapped, but not to y, then y is considered wrongly assigned and it is replaced by the set of tags that are mapped to x in M_c.

The corpus is now brought to a form where each token may have one or more tags assigned. Let this version be denoted by A*(Y), and called the star version of the corpus A, tagged with the tagset Y. In the next section we show how we selected one single tag for the tokens having assigned more than one in the star versions of the corpora.

4.2. Choosing the Right Tag

4.2.1. The Algorithm

The tag selection is done by retagging. Again the procedure is independent for the two corpora and we will describe it only for one of them.

The retagging process is stochastic, based on trigrams. The language model is learnt from the gold standard. We build a Markov Model that has bigrams as states and that emits tokens each time it leaves a state. To find the most likely path through the states of the Markov Model, we used the Viterbi algorithm, with the restriction that the only tags available for a token are those assigned to that token in the star version of the corpus. That means that at any moment only a limited number of states are available to chose from.

4.2.2. Lexical Probabilities

The lexical probabilities involved in the Viterbi algorithm have the form $p(w_k|x_i)$, where w_k is a token and x_i a tag. For $<w_k,x_i>$ pairs unseen in the training data, the most likelihood estimation procedure would assign null probabilities ($p(w_k,x_i)=0$ and therefore $p(w_k|x_i)=0$).

We smoothed the $p(w_k,x_i)$ probabilities using the Good-Turing estimation, as described in (Gale & Sampson, 1995).

The probability mass reserved for the unseen token-tag pairs (let it be p_0) must somehow be distributed among these pairs. We constructed the set UTT of all unseen token-tag pairs. Let T(x) be the number of token types tagged x. The probability p(w,x), with $<w,x>\in$UTT, that a token w might be tagged with the tag x was considered to be directly proportional with T(x), that is:

$$p(w, x) / T(x) = u = \text{constant}$$

Now p_0 can be written as follows:

$$p_0 = \sum_k \sum_i p(w_k, x_i) \text{, where } <w_k,x_i> \in \text{UTT}$$

In UTT all N(x) pairs of the type $\{<w_1,x>, <w_2,x> \ldots <w_{N(x)},x>\}$ are considered of equal probability, u*T(x). It follows that:

$$p_0 = \sum_i N(x_i) \cdot u \cdot T(x_i) = u \cdot \sum_i N(x_i) \cdot T(x_i)$$

The lexical probabilities for unseen token-tag pairs can now be written as:

$$p(w, x_i) = \frac{p_0 \cdot T(x_i)}{\sum_i N(x_i) \cdot T(x_i)} \text{ for any } <w,x_i> \in \text{UTT}$$

4.2.3. Contextual Probabilities

The contextual probabilities are obtained by linear interpolation of unigram, bigram, and trigram probabilities, that is:

$$p(x_i|x_1,\ldots,x_{i-1}) = \lambda_1 p(x_i) + \lambda_2 p(x_i|x_{i-1}) + \lambda_3 p(x_i|x_{i-2},x_{i-1})$$

where $\lambda_1 + \lambda_2 + \lambda_3 = 1$.

We estimated the values for the coefficients for each combination of unigram, bigram and trigram, met while traversing the corpus. As a general rule, we considered that the greater the observed frequency of an n-gram and the fewer (n+1)-grams beginning with that n-gram, the more reliable such an (n+1)-gram is.

We first estimated λ_3. Let $F(x_{i-2},x_{i-1})$ be the number of occurrences for the bigram $x_{i-2}x_{i-1}$ in the training data. Let $N_3(x_{i-2},x_{i-1})$ be the number of distinct trigrams beginning with that bigram. Then the average number of occurrences for a trigram beginning with x_{i-2},x_{i-1} is:

$$F_3(x_{i-2}, x_{i-1}, \bullet) = F(x_{i-2}, x_{i-1}) / N_3(x_{i-2}, x_{i-1})$$

Let $F_{3max} = \max_{x_{i-2},x_{i-1},\bullet} F_3(x_{i-2},x_{i-1},\bullet)$. We took λ_3 to be:

$$\lambda_3 \triangleq \log(F_3(x_{i-2}, x_{i-1})) / \log(F_{3max})$$

Similarly λ_2 is computed as:

$$\lambda_2 = (1 - \lambda_3) \log(F_2(x_{i-1})) / \log(F_{2max})$$

and $\lambda_1 = 1-\lambda_2-\lambda_3$.

We have now completely defined the retagging algorithm and with it, the whole cross-tagging method. Does it improve the performance of the direct tagging? Our experiments show it does.

5. Experiments and Evaluation

5.1. Resources

We used two English language corpora as gold standards. The 1984 corpus, having approximately 120,000 tokens, contains the George Orwell's homonymous novel. It was automatically tagged, but it was thoroughly human validated and corrected. The tagset used in this corpus is called the MTE tagset.

The second corpus was a fragment of the tagged SemCor corpus, of about the same length, referred to as SemCorP (partial). The full SemCor corpus, is 778,587 tokens in length and was created by the Princeton University and is distributed with the Princeton WordNet. The texts included into SemCor represent a subset of the Brown corpus (http://khnt.hit.uib.no/icame/manuals/brown/INDEX.HTM), compiled in the '60s by W. Francis and H. Kucera of Brown University. The SemCor corpus was tagged with the Brill tagger and uses the Penn tagset (http://multisemcor.itc.it/semcor.php). In the SemCor corpus only the word forms are POS-tagged, therefore we assigned a different tag for each type of punctuation mark. As tagger of reference for our tagging experiments, we used the TnT tagger developed by T. Brants (Brants, 2000).

5.2. Experiment 1

After cross-tagging the two corpora described above, we compared the results with the direct-tagged versions: 1984_{DT}(Penn) against 1984_{CT}(Penn) and $SemCorP_{DT}$(MTE) against $SemCorP_{CT}$(MTE). There were 6,391 differences for the 1984 corpus and 11,006 for the SemCorP corpus. As we did not have human-validated versions of the two corpora, tagged with each other's tagset, we randomly selected a sample of one hundred differences for each corpus and manually analyzed them. The result of this analysis is shown in Table 7.

	Correct CTtags	Correct DTtags
100 differences in 1984(Penn)	69	31
100 differences in SemCorP(MTE)	59	41

Table 7. Cross-tagging results

Overall, cross-tagging is shown to be more accurate than direct tagging. However, as one can see from Table 7, the accuracy gain is more significant for the 1984 corpus than for SemCorP. The explanation is the following:

Since the language model built from the 1984 corpus (used for direct tagging of SemCorP) is more accurate than the language model built from SemCorP (used for direct tagging of 1984), there were much more errors in 1984(Penn) than in SemCorP(MTE). The cross-tagging approach described in this paper has the ability to overcome some inconsistencies encoded into the supporting language models.

5.3. Experiment 2

We decided to improve the POS-tagging of the entire SemCor corpus. First, to keep track of the improvements of the corpus annotation, we computed the identity score between the original and the biased-tagged versions. Let S^0(Penn) be the SemCor corpus in its original form, and S^0_{BT}(Penn) its biased tagged version.

$$\text{Identity-score}(S^0(\text{Penn}), S^0_{BT}(\text{Penn})) = 93.81\%$$

By cross-tagging the results of the first experiment, we obtained the double cross-tagged version of SemCor(Penn) which we denote as S^1(Penn).

$$\text{Identity-score}(S^0(\text{Penn}), S^1(\text{Penn})) = 96.4\%$$

These scores were unexpectedly low and after a brief analysis we observed some tokenization inconsistencies in the original SemCor, which we normalized. For instance, opening and closing double quotes were not systematically distinguished (their number was equal); in this case we turned all the instances of `` and '' into the DBLQ character ("). Another example of inconsistency refered to various formulae denoted in SemCor some times by one token **f and some other times by a sequence of three tokens *, *, f; in the normalized version of the SemCor only the first type of tokenization was preserved. Let S^2(Penn) denote the normalized version of S^1(Penn).

$$\text{Identity-score}(S^2(\text{Penn}), S^2_{BT}(\text{Penn})) = 97.41\%$$

As one can see, the double cross-tagging and the normalization process resulted in a more consistent language model (the biased-tagging identity score improved with 3.6%).

At this point, we analyzed the tokens that introduce the most differences. For each such token, we identified the patterns corresponding to each of their tags and subsequently corrected the tagging to match these patterns. The tokens considered in this stage were: *am*, *are*, *is*, *was*, *were*, and *that*. Let S^3 be this new version of the corpus.

$$\text{Identity-score}(S^3(\text{Penn}), S^3_{BT}(\text{Penn})) = 97.61\%$$

Finally, analyzing the remaining differences, we notices very frequent errors in tagging the grammatical number for nouns and mistagging common nouns as

proper nouns and vice versa. We used regular expressions to make the necessary corrections and thus obtained a new version S^4(Penn) of SemCor.

Identity-score(S^4(Penn), S^4_{BT}(Penn)) = 98.08%

Continuing the biased correction/evaluation cycle would probably further slightly improve the identity score, but the distinction between correct and wrong tags becomes less and less clear-cut. The overall improvement of the biased evaluation score (4.27%) and the observed difference types suggested that the POS tagging of the SemCor corpus reached a reasonable level of accuracy for making it a reliable training corpus.

To assess the improvements in S^4(Penn) over the normalized version of the initial SemCor corpus we extracted the differences among the two versions. The 57,905 differences, were frequency-based sorted and resulted 10216 difference types, with frequencies ranging from 1910 to 1. The 10 most frequent difference types are shown in Table 8.

Double Cross-Tagging Tag	Token	Original Tag	Frequency
TO	to	VB	1910
VBN	been	VB	674
IN	in	RB	655
IN	in	VB	646
IN	of	RB	478
IN	on	VB	381
IN	for	VB	334
IN	with	VB	324
RBR	more	RB	314
DT	the	RB	306

Table 8. The first 10 most frequent differences between the double-cross tagging and the original tagging in SemCor

The first 200 types, with frequencies ranging from 1910 to 40 and accounting for 25136 differences, were thoroughly evaluated. The results of this evaluation are shown in the Table 9.

# differences	Double Cross-Tagging OK	Original tagging OK
25136	21224 (84.44%)	3912 (15.56%)

Table 9. Analysis for the most frequent 200 difference types among the initial and final versions of the SemCor corpus

6. Conclusions

In the light of the performed experiments, the cross-tagging showed useful for several purposes. The direct tagging of a corpus can be improved. Two tagsets can be compared from a distributional point of view. Errors in the training data may be spotted and corrected. Successively applying the method for different pairs of corpora tagged with different tagsets would permit the building of a much larger corpus, tagged in parallel with all those tagsets in a reliable manner.

The mapping system between two tagsets may prove useful in itself. It is composed of a global map, as well as of many token maps, showing the way contexts marked by certain tags in one tagset overlap with contexts marked by tags of the other tagset. Furthermore, the mapping system can be applied not only to POS tags, but to other types of tags as well.

It is to be expected that using a better tagger or more accurately tagged gold standards, the mapping system will improve and hence the overall cross-tagging performance.

7. References

Brants, Thorsten (2000). TnT – A Statistical Part-of-Speech Tagger. *In Proceedings of the 6th Applied NLP Conference*. Seattle, WA, pp. 224-231.

Brants, Thorsten (1995). Tagset Reduction Without Information Loss. I*n Proceedings of the 33rd Annual Meeting of the ACL*. Cambridge, MA, pp. 287-289.

Chen, Stanley F. & Goodman, Joshua (1996). An Empirical Study of Smoothing Techniques for Language Modeling. *In Proceedings of the 34th Annual Meeting of the ACL*. Santa Cruz, CA, pp. 310-318.

Church, Kenneth W. (1988). A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. *In 2nd Conference on Applied NLP*. Austin, TX, pp. 136-143.

Gale, William A. & Sampson, Geoffrey (1995). Good-Turing Frequency Estimation Without Tears. *In Journal of Quantitative Linguistics*, 2/3, pp. 217-237.

Manning, Christopher D. & Schütze, Hinrich (1999). Foundations of Statistical Natural Language Processing, The MIT Press, Cambridge, MA, London, England.

Tufiş, Dan & Dragomirescu, Liviu (2004). Tiered Tagging Revisited. *In Proceedings of the 4th LREC Conference*. Lisbon, Portugal, pp. 39-42.