LARGE TAGSETS AND HIGH ACCURACY IN STATISTICAL MORPHO-SYNTACTIC DISAMBIGUATION OF WRITTEN TEXTS

Dan TUFIŞ

RACAI-Romanian Academy
13, 13 Septembrie St., 74311 Bucharest, Romania
E-mail: tufis@valhalla.racai.ro

Abstract. It is well known that for inflectional languages large tagsets are required in order to adequately describe the content of a word-form lexicon. According to the EAGLES/MULTEXT recommendations for standardised encoding of dictionaries, further refined by the MULTEXT-EAST specifications, for such languages several hundreds and even thousands of tags are possible. However, because of various limitations of the supervised learning procedures and of the n-grams language modelling such a large tagset can hardly be used for a reliable disambiguation process. We present a novel approach, based on statistical (supervised) learning and processing, aiming to ensure the benefits of the statistical processing but also to reconcile the information richness of a large tagset and the high accuracy of the morpho-syntactic disambiguation as required by all modern applications of language technology.

1. INTRODUCTION

The linguistic problem addressed here is the morpho-syntactical disambiguation (MS-tagging) of highly inflectional natural language arbitrary texts. This can be regarded as a classification problem: an ambiguous lexical item is the one that can in different contexts be classified differently and given a specified context the disambiguator/classifier decides on the appropriate class. The features that are relevant to the classification task are encoded into the tags. Given that not all lexical features are equally good predictors for distributional properties of a language, usually the tags use only a subset of the features encoded into a morpho-syntactic lexicon.

It is generally accepted that the state-of-the-art in MS-tagging makes room for significant improvements of accuracy. The granularity of the tagset will

determine the difficulty and the complexity of the MS-tagging task. If the basic language model (LM) distinguished only a few categories of linguistic units, each of them with a small number of attributes, then the cardinality of the necessary tagset would be small and the size of the necessary training data would be easily attainable. On the contrary, if the LM distinguished among several classes of linguistic units and these were described in terms of a larger set of attributes, the necessary tagset would be inherently higher than in the previous case. The larger the tagset, the larger the training corpora needed [3]. Moreover, having to choose among an increased number of possibilities, it seems quite intuitive that the MS-tagging becomes harder as the granularity of the LM gets finer.

There is therefore a tension between the number of feature combinations that are relevant and noiseless in statistical analysis (corpus tags, or Ctags) and the number of linguistically motivated feature combinations defined in a lexicon (morpho-syntactic descriptors, or MSDs).

Our approach, called tiered tagging, is one possible way to reconcile probabilistic tagging techniques (based on corpus tags) with the information richness provided by lexicon MSDs. With a very small price in tagging accuracy (as compared to a direct Ctags approach), and practically no price in computational resources, tiered tagging ensures disambiguation of a text in terms of MSDs by using LMs built for Ctags. Tiered tagging uses the Ctags as a hidden tagset, for a first level of tagging. Then a post-processor deterministically replaces the Ctags by one or more MSDs (in our experiments never more than 3). The words that this replacement makes ambiguous (in terms of the MSD-tagset annotation) are more often than not the difficult cases in statistical disambiguation. Very simple contextual rules (regular expressions) differentiate the interpretations of the few still ambiguous words (in our experiment, less than 10%). Because the application of contextual rules is rarely required, the response time penalty is insignificant. With an estimated error rate for rule-based disambiguation below 1.5%, the overall error rate due to the second phase of the tiered tagging process is less than 0.15%(1.5%*10%), or put it otherwise, if the first phase would be errorless, the overall accuracy of the tiered tagging would be higher than 99.85%!

Obviously, the Ctags and the MSDs have to be in a specific relation and the appropriate design of the reduced tagset is crucial. In [13] it is shown how an initial reduced tagset could be interactively designed from a large tagset (MSD set), based on a trial-and-error ID3-like procedure (the resulted reduced tagset subsumed the larger one). Then, by observing the systematic errors made by the tagger, the Ctags were further generalised to get rid of the features responsible for mistagging. This generalization process was the

result of an introspective analysis but supported by evidence provided by the corpus data. The procedure was repeated several times until the error rate stabilised at an acceptable level (around 97.5%). Although this trial-end-error tagset design process took almost one man-year work, the final results were awarding.

Another important issue we were concerned with was finding whether it could be possible to improve the tagging accuracy by combining register diversified LMs. This was a real challenge, as the general practice in statistical tagging is to use balanced training corpora, as large as possible. We found that combining register diversified LMs was worth doing and besides a higher accuracy than in the traditional approaches, there was an additional bonus which came for free, namely a hint on the register of the new tagged text. Although we did not follow this path, there are some new investigations exactly in this spirit ([2]).

2. MORPHO-SYNTACTIC DESCRIPTORS, CORPUS TAGS, AND TIERED TAGGING

In order to promote multilinguality, one of the important issues is standardisation of language resources encoding. In the area of morpho-syntactic information encoding, one of the most influential standard proposals is the EAGLES (Expert Advisory Group on Language Engineering Standards) report "Synopsis and comparison of morpho-syntactic phenomena encoded in lexicons and corpora. A common proposal and applications to European languages" (see http://www.ilc.pi.cnr.it/EAGLES/home.html). The recommendations of EAGLES have been adopted by the MULTEXT project (among the others) and further extended by the MULTEXT-EAST project to cover new 6 Eastern European languages (http://nl.ijs.si/ME). As partners of the MULTEXT-EAST project, we developed various linguistics resources compliant with the EAGLES recommendations. These recommendations, that took into account the majority of European languages, identified a set of 14 part-of-speech categories, and for each of them a large set of features (some common for all languages, some language specific). Then, for each feature of every part-of-speech, the recommendations list all possible values. Part-of-speech categories, features and values are encoded by using one character identifier. The proper morpho-syntactic descriptors are provided as strings using linear encoding. In this notation, the position in a string corresponds to a feature and specific characters in each position indicate the value for the corresponding feature. That is, the positions in a string of characters are numbered 0, 1, 2, etc. and are used in the following way:

- the character at position 0 encodes the part-of-speech;
- each character at position 1, 2,..., n encodes, using one character code, the value of one feature (person, gender, number, etc.) for the part-of-speech specified by the character at position 0;
- if an feature does not apply, the corresponding position in the string contains a special marker '-' (hyphen).

The "does-not-apply" marker in the MSD encoding needs a few explanations. Besides the basic meaning that the feature is not valid for the language in case, it can also indicate that a certain combination of feature values makes the current feature irrelevant (e.g., Person feature in case of non-finite verbal forms). The EAGLES recommendations provide another special feature value, the dot ("."), for cases where an attribute can take any value in its domain. The "any" value is particularly useful in dealing with under-specified encodings as needed for instance by the representation of syncretism. However, because MULTEXT-EAST recommendations did not included the "dotted" notation, it was decided to represent under-specified values also by means of '-' marker. Following the lexicon encoding specifications we constructed a wide coverage word-form lexicon for Romanian, each entry having the format WORD-FORM LEMMA MSD.

The homographs are represented as different entries, differentiated by the MSD (and sometimes by lemma). As usual, all the MSDs that are associated with one word-form constitutes the MSD-ambiguity class of that wordform. Table 1 provides information on the data content of the main Romanian lexicon used for the corpus analysis. A full account of the dictionary encoding strategies, its content and various statistics can be found in [15].

Table 1 – Romanian dictionary overview

Entries	Items	Lemmas	MSDs	AMB-MSD
421169	347252	36028	614	869

AMB-MSD represents the number of ambiguity classes. Based on this set of morpho-syntactic descriptors (MSDs), and for tagging purposes, we designed a reduced tagset (Ctags tagset) containing 82 tags (plus 10 punctuation tags). The reduced tagset, obtained by a trial&error process, eliminated attributes or merged attribute values which were either distributionally irrelevant or fully lexicon recoverable based on the remaining attribute values. Yet some attributes and values, although fully recoverable, were preserved in the reduced tagset, because they help disambiguate the surrounding words. The main

property of this reduced tagset is what we call *recoverability*, to be described as follows.

Let MAP: $Ctagset \rightarrow MSDset^m$ be a function that maps a Ctag onto an ordered set of MSDs, AMB: $W \rightarrow MSDset^n$, a function that maps a word onto its ambiguity class (from the lexicon) and TAG: $W \rightarrow Ctagset$, a selector that returns for a word the tag assigned by a tagger (in a specific context). Then, recoverability (as achieved in our tagset design) means:

$$\text{card}(AMB(w) \cap MAP(TAG(w))) = \begin{cases} 1, & \text{in more than 90\% cases,} \\ \geq 2, & \text{for less than 10\% cases.} \end{cases}$$

The reduced tagset has the property that one tag assigned to a given word w can be deterministically mapped back onto the appropriate MSD in the large tagset in more than 90% of the cases. Note that although this mapping is almost deterministic, one tag may be mapped differently, depending on the context and the word it is assigned to.

In our experiments we used two different 3-gram HMM-based taggers working in quite a similar manner in the training/learning phase. The difference among them consisted in the optimization technique they used (local vs. global) and in the guessing approach (guesser automatically constructed from the training data vs. morphology-based hand written guesser). The training process was conducted under identical conditions for the two taggers. Based on George Orwell's "1984", Plato's "The Republic" and several issues from "România Liberă" and "Adevărul" (the newspapers with the largest distribution in Romania), we constructed three different register training corpora (fiction, philosophy and journalism). The training corpora were hand-disambiguated in terms of the reduced tagset (Ctagset) and they served for building several language models (LM), one for each training corpus and each tagger.

Table 2 – Romanian training corpora overview

Corpus	Occurrences	Items	Lemmas	MSDs	AMB-MSD
1984	118356	16142	9318	414	589
Republic	135349	12859	7842	398	520
News	92669	9673	5944	403	416
Global	346374	25676	13386	581	677

An overview of these texts is given in Table 2. The three training corpora were concatenated and used in the generation of the baseline LM (referred to in the following as the Global). For testing purposes, we extra hand-tagged

about 60,000 words from different texts in the three registers: **Fiction** ("1994" ([11]) – a follow-up story of Orwell's famous novel), **Philosophy** ("Aristotle" ([1]) – a study of Aristotle's work) and **Newspapers** (a collection of articles from newspapers others than those included in the training corpora).

The morpho-syntactic disambiguation of a new text (unseen in the training phase) is done in two consecutive steps (this is why we call this approach *tiered tagging*):

- first, the text is tagged by a classifier, constructed in the training phase, which assigns a Ctag for each item in the input text;
- the second step, achieves the mapping of the Ctags into MSDs by simply computing the intersection

$$(AMB(w) \cap MAP(TAG(w))$$

and assigning the resulted set of MSDs to the current token.

As we said before, the Ctagset was designed so that the mapping would be in the vast majority of cases deterministic, that is the before-mentioned intersection would contain only one MSD. For the rare cases when a coarse-grained tag is not mapped onto a unique MSD but onto a list of MSDs, we use 14 very simple contextual rules (regular expressions). They specify, by means of relative offsets, the local restrictions made on the current tag assignment. Our rules inspect the left, the right or both contexts with a maximum span of 4 words. Such a rule, headed by a list representing the still there ambiguity, is a sequence of pairs (*MSD: conditions*) where *conditions* is a disjunction of regular expressions which, if applied to the surrounding tokens (defined as positive or negative offsets), returns a truth-value. If *true*, then the current token is assigned the *MSD*, otherwise the next pair is tried. If no one of the conditions returns a *true* value, the mapping ambiguity remains unsolved. This happens very rarely (for less than 1% of the whole text). For instance, the following rule considers a tag class DS corresponding to two merged MSD classes (possessive pronouns and possessive determiners/adjectives).

$$
\begin{aligned}
&Ps|Ds\\
&Ds.\alpha\beta\gamma:\ (-1Nc\alpha\beta\gamma y)||(-1Af.\alpha\beta\gamma y)||(-1Mo.\alpha\beta\gamma y)||\\
&\quad(-2Af.\alpha\beta\gamma n \text{ and } -1Ts)||(-2Nc\alpha\beta\gamma n \text{ and } -1Ts)||\\
&\quad(-2Np \text{ and } -1Ts)||(-2D..\alpha\beta\gamma \text{ and } -1Ts),\\
&Ps.\alpha\beta\gamma:\ true.
\end{aligned}
$$

The rule reads as follows (α, β, γ represent shared attribute values, "." represent an "any" value):

IF any of the conditions a) to g) is true
a) previous word is a definite common noun
b) previous word is a definite adjective
c) previous word is a definite ordinal numeral
d) previous words are an indefinite adjective followed by a possessive article
e) previous words are an indefinite common noun followed by a possessive article
f) previous words are an indefinite proper noun followed by a possessive article
g) previous words are a determiner followed by a possessive article,
THEN choose the determiner MSD; shared attribute values set by the context
ELSE choose the pronominal MSD; shared attribute values set by the context.

3. COMBINED CLASSIFIERS METHODS

In very general terms, a classifier is a function that, given an input example, assigns it to one of the k classes. In contrast, a learning algorithm is a function that, given a set of examples and their classification, constructs a classifier ([8]). When trying to combine different classifiers one would certainly prefer classifiers of comparative accuracy and more important classifiers that would not make identical errors. The basic idea in combining classifiers is to complement each other's decisions so that to minimise the number of errors. There are different statistical tests to check these hypotheses, out of which we used McNemar and Brill&Wu's tests.

3.1. COMBINING TAGGERS VERSUS COMBINING LANGUAGE MODELS

Recent work on combined classifier methods ([7], [10], [5]) has shown one effective way to significantly speeding up the process of building high quality training-corpora with a corresponding costs diminishing. The combined taggers approach for MS-tagging ([10], [5]) is intuitively described below.

Having k different MS-tagging systems and a training corpus, build k LMs, one model for each system. Then, given a new text T, run each trained tagging system on it and get k disambiguated versions of T, namely $T_1, T_2, \ldots, T_k$. Put it otherwise, each token in T would be assigned k interpretations (not necessarily distinct). Given that each tagging system has its own idea of the processed text (encoded in its associated LM), it is very unlikely that the k versions of T shall be identical. However, as compared to *truth* (a human

judged annotation), the probability that an arbitrary token from T is assigned the correct interpretation in at least one of the k versions of T is very high (more than 99%). Let us call the hypothetical guesser of this correct tag an *oracle* (as called in [5]). Implementing an oracle, i.e,. automatically deciding which of the k interpretations is the correct one is hard to do. However, the oracle concept, as defined above, is very useful since its accuracy allows an estimation of the upper bound of correctness that can be reached by a given taggers combination.

The experiment described in [10] is based on the tagged LOB corpus and uses four different taggers: a trigram HMM tagger(the TOSCA tagger), a memory-based tagger ([6]), a rule-based tagger ([4]) and a Maximum Entropy-based tagger ([12]). Several decision-making procedures are proposed and with a pair-wise voting strategy, the combined classifier system outscored (97.92%) all the individual tagging systems. However, the oracle's accuracy for this experiment (99.22%) proves that investigations on the decision-making procedure should go on.

An almost identical point of view is shared and similar results are reported in [5]. Their experiment is based on the Penn Treebank Wall Street Journal corpus and uses a HMM trigram tagger, a rule-based tagger ([4]) and a Maximum Entropy-based tagger ([12]). Here, the expected accuracy of the oracle is 98.59%, and using the "pick-up tagger" combination method, the overall accuracy was 97.2%.

It is very interesting that both experiments show that different taggers, based on LMs constructed from the same training data, make complementary errors, and this happens both when considering two taggers that perform equally well and when considering a sophisticated tagger (such a maximum-entropy tagger) and a very simple one (such as a unigram tagger). Therefore, it *does* make sense to look for combination methods.

At first sight our proposal looks alike, but in fact it is quite different. In our experiment we used only one tagger T, but it was trained on different register texts. We buit this way different language models $(LM_1, LM_2, \ldots, LM_k)$, the difference among them being justified only by linguistic data and not by the learning algorithm. A new text (unseen, from an unknown register) is independently tagged with all available classifiers (as in the "multiple taggers" approach) and their outputs are combined for the final result.

We made experiments with various combiners (simple majority, weighted majority, etc.). The best performing one is called *CREDIBILITY* and seems to be quite similar to the *pick-up tagger* combiner used in ([5]). This combiner is driven by a set of *credibility profiles* (one for each classifier). A credibility profile, automatically constructed from evaluating a classifier on the *Global*

training corpus (see Table 2), specifies for each tag T_i the probability of its correct assignment $Pr(T_i)$ by the classifier in case and a confusion set. The confusion set for a tag T_i is a list of pairs $\langle T_i P_c^k(T_i|T_j)\rangle$, with T_j a tag that is wrongly used instead of T_i and $P_c^k(T_i|T_j)$ the probability of such a confusion. The following relation describes the *CREDIBILITY* combiner:

$$\text{argmax}_k C^k(T_i) = Pr^k(T_i) - \sum P_c^k(T_i|T_j)^* \beta(T_j)(j \neq i),$$

where $C^k(T_i)$ is the credibility that the k-th classifier is right and

$$\beta(T_j) = \begin{cases} 1, & \text{if } T_j \text{ is assigned by another classifier,} \\ 0, & \text{otherwise.} \end{cases}$$

The winning tag is the one proposed by the classifier with the highest credibility.

While in the "multiple taggers" approach it is very hard to see the influence of the type of training data, in our "multiple registers" approach, one could get a strong indication on the type of the currently processed text. As our experiments have shown, when a new text belonged to a specific language register, that language register model never failed to provide a higher accuracy in tagging than any other specific register language model. Having a general hint on what type of a text the one currently processed is, then stronger identification criteria could be used to validate the hypothesis.

4. EVALUATION, AVAILABILITY, AND CONCLUSIONS

The evaluation discussed in this section refers to the hidden layer of the tiered tagging, because the MSD recovering is done after the combination of the individual classifiers. As we said before, the MSD recovering introduces a distortion of the hidden layer tagging accuracy of less than 0.15%.

In Section 2 we mentioned that the tagger used for the combined language model (CLAM) approach was a 3-gram HMM tagger. Initially, this was a slightly modified version of O. Mason's QTAG trigram tagger. QTAG uses a local optimization strategy, a sliding 3-word window with the word of interest in the 1^{st}, 2^{nd} and 3^{rd} position respectively. Being interested in verifying that the C-tagset designed was not over-tuned and biased by the peculiarities of QTAG, we have repeated the same experiments with another 3-gram tagger, namely TnT due to T. Brants (available from the author, licence-based). TnT uses the same input/output format but, unlike QTAG, the optimal sequence of tags is globally computed (Viterbi algorithm). The evaluation showed different results for the two taggers (and we will discuss them further), but, in both

cases, our basic claim was confirmed: *TT-CLAM methodology ensures high accuracy in tagging with a large tagset.*

Table 3 shows the results of the evaluation process for 8 single classifiers and 2 combined classifiers (CLAM), built based on the two taggers (QTAG and TnT) and four training corpora (Orwell's "1984", Plato's "The Republic", News and Global). They were run on three texts unseen before, representing chunks of approximately 20,000 words from the previously mentioned 3 test corpora). The test texts contained unknown words.

Table 3 – Classifiers evaluation; the test texts contain unknown words

				QTAG		TnT	
Text/LM	Size	Amb	Unk	Accuracy	#errs	Accuracy	#errs
1994_20/CLAM	20110	1.54	26	98.42	318	98.45	313
1994_20/Global				98.32	338	98.20	361
1994_20/1984				98.23	356	98.08	385
1994_20/News				97.88	425	97.87	427
1994_20/Rep				97.76	450	97.84	433
				Avg.=98.12		Avg.=98.08	
barnes_20/CLAM	20120	1.58	158	97.06	590	97.45	512
barnes_20/Global				96.92	620	97.15	572
barnes_20/News				96.89	624	96.96	610
barnes_20/Rep				96.62	680	96.95	613
barnes_20/1984				96.56	692	96.92	619
				Avg.=96.81		Avg.=97.08	
ziarNow_20/CLAM	20035	1.57	248	97.36	527	98.33	336
ziarNow_20/News				97.34	533	98.18	365
ziarNow_20/Global				97.18	564	98.30	342
ziarNow_20/Rep				96.73	655	97.94	414
ziarNow_20/1984				96.50	701	97.87	427
				Avg.=97.02		Avg.=98.14	

The column *Text/LM* specifies the chunk of the test text that was tagged by using a specific LM. The *Size* column specifies the number of lexical tokens in the test text. The *Amb* describes the average ambiguity of the tagged text, computed as the number of tags before disambiguation divided by the number of tokens. The ambiguity is computed for the hidden layer of tags (Ctagset). If one considers the ambiguity in terms of MSDs (final delivery annotation) and disregard unambiguous tokens (punctuation, numbers, etc), then the *Amb* figures in Tables 3 and 4 are well above 2.5 (2.59, 2.75, 2.68). The *#Unk* column specifies the number of unknown items in the test text. The *Accuracy* and *#errs* columns describe, for each tagger that was used in building a classifier, the percentage of correct tag assignment (number of correctly assigned tags versus the number of tags) and the absolute number

of errors, respectively. For instance, the chunk "1994_20", containing 20110 items, when tagged with a classifier based on the "News" LM (1994_20News) constructed with TnT, contained 427 errors, thus its accuracy was 97.87%. As one can see, the CLAM classifier was always the best performer, irrespective of the used tagger, language models or the test text.

Table 3 also shows that the classifiers based on the *Global* LM were almost always the second-best ones (except for the text *ziarNou_20*, when they were in the third position). This supports the already known fact that more training data improve the tagging performance but also sustains our conjecture: *dividing a balanced training corpus into register-specific training corpora and using a combined LM classifier could further increase the tagging performance.*

It is worth noting that after we analysed the errors, we found out that the non-Romanian word "qua" occurred in "barnes_20" 81 times, used as a close-class category (conjunction). Similarly, the text "ziarNou" contained 43 occurrences of the item "lu" (used, for stylistic purposes, as a slang form of the pronoun "lui"). The guesser used in QTAG, although morphology based and tuned for Romanian, considered, as the current practice is, only open-class tags so, it always failed to assign a correct ambiguity class for both words, and consequently these two words were always mistagged. By error propagation, about 150 errors in "barnes_20" and about 70 errors in "ziarNou" were directly attributable to these two items. The guesser used in TnT was automatically constructed based on the suffixes of the items seen in the training corpora. Therefore, it managed to correctly tag 27 occurrences of "qua" and 12 occurrences of "lu". The figures in Table 3 show a better performance for the classifiers constructed with TnT. However, when the two anomalous words were normalised ("*qua*" and "*lu*" were defined as aliases for the conjunction "ca" and the pronoun "lui", respectively), the QTAG-based classifiers were practically as good as the TnT-based classifiers.

In a second experiment, we introduced all the previously unknown items in the dictionary, all the possible interpretations being assigned equal lexical probabilities. Table 4 displays the results of this second experiment.

Without unknown items in the test text, when comparing the performances of the classifiers trained on the same corpus, the difference in accuracy could be explained by the different optimization techniques used by the two taggers. The experiment supported the idea that global optimization is in general better than the local one, unless too many unknown words are present in the input text.

An estimate (see Table 5) of the effect of a wrong guess on tagging overall accuracy with respect to the global/local optimization strategy may be given by the value $\mu = (N_{PL} - N_{FL})/N_{PL}$, where N_{PL} = the number of wron-

gly assigned tags in a text containing unknown words and N_{FL}=the number of wrongly assigned tags in the same text, but with the previously unknown words included into the lexicon. Although the percentage of unknown words recovery is better for TnT than for QTAG with all the classifiers, μ_{TnT} is always greater than μ_{QTAG}, implying that the effect of a wrong guess is reduced in local optimization approach. This is even more obvious if one considers the normalisation of the two items "qua" and "lu": the $\%_{QTAG}$ becomes slightly better than $\%_{TnT}$, but the ratio μ_{TnT}/μ_{QTAG} becomes even greater than before.

Table 4 – Classifiers evaluation; the test texts contain no unknown words

				QTAG		TnT	
Text/LM	Size	Amb	Unk	Accuracy	#errs	Accuracy	#errs
1994_20/CLAM	20110	1.54	0	98.71	260	98.74	254
1994_20/Global				98.69	264	98.56	289
1994_20/1984				98.54	290	98.45	310
1994_20/News				97.15	374	98.36	329
1994_20/Rep				98.29	342	98.30	341
				Avg.=98.47		Avg.=98.47	
barnes_20/CLAM	20120	1.58	0	99.00	203	99.11	180
barnes_20/Global				98.64	275	98.93	215
barnes_20/News				98.57	289	98.80	241
barnes_20/Rep				98.31	340	98.73	254
barnes_20/1984				98.43	316	98.67	266
				Avg.=98.59		Avg.=98.85	
ziarNow_20/CLAM	20035	1.57	0	98.88	225	99.20	160
ziarNow_20/Global				98.30	341	99.14	172
ziarNow_20/News				98.13	376	99.05	192
ziarNow_20/Rep				97.77	447	98.93	213
ziarNow_20/1984				97.57	488	98.78	244
				Avg.=98.13		Avg.=99.02	

Our experiments, intensive tests and evaluations with various classifiers brought evidence for several challenging hypotheses which we believe are language independent:

- the *error-complementarity* conjecture holds true for the LMs combination. We tested this conjecture with 56 classifier combinations on various texts (about 20.000 words each) in three different registers (fiction, philosophy and journalism) and no experiment contradicted it;
- a text T_i belonging to a specific register R_i is more accurately tagged with the LM_i learnt for that register than if using any other LM_j. As a consequence, by tracking which one of the (LM-dependent) classifiers

came closer to the final tag assignment, one could get strong evidence for text-type/register identification (with the traditional methods further applicable);

- the combined LMs classifier method does not depend on a specific tagger. The better the tagger, the better the final results.

- splitting a balanced training corpora into specialised registered training corpora is worth considering: even a simple combiner as MAJORITY ensures a better result than using only the LM of the initial balanced corpus;

- the high level of correct agreement and the negligible percentage of false agreement can help in fast and cheap development of large training corpora. The human expert annotator can concentrate quite safely on the disagreement cases. With a less than 2.5% (in our experiments), the hand disambiguation of large training corpora is a manageable task. In our experiments we haven't observed any instance of disagreement between the 4 classifiers were the right tag was not proposed by at least one of them.

Table 5 – The interactions between guessing errors and optimization technique

		TnT					QTAG				
Text/LM	Unk	OK	%	N_{PL}	N_{FL}	μ	OK	%	N_{PL}	N_{FL}	μ
1994_20/1984	26	18	0.69	385	310	0.195	16	0.61	356	290	0.18
1994_20/Rep		18	0.69	427	341	0.20	17	0.65	425	342	0.19
1994_20/News		15	0.57	433	329	0.24	16	0.61	450	374	0.16
Barnes_20/1984	158	104	0.65	613	266	0.56	66	0.41	680	316	0.53
Barnes_20/Rep		94	0.59	619	254	0.59	65	0.41	692	340	0.50
Barnes_20/News		101	0.63	610	241	0.60	70	0.44	624	289	0.53
ziarNow_20/1984	248	191	0.77	427	244	0.42	130	0.52	701	488	0.30
ziarNow_20/Rep		181	0.72	414	213	0.48	131	0.52	655	447	0.31
ziarNow_20/News		196	0.79	336	172	0.48	141	0.56	527	341	0.35

Acknowledgements: The work reported here built on the main results of the Multext-East(COP106/1995) and TELRI(COP200/1995) European projects and was partly funded by a grant of the Romanian Academy (GAR188/1998).

REFERENCES

[1] J. Barnes, *Aristotel*, Editura Humanitas, Bucureşti, 1992.

[2] D. Beeferman, A. Berger, J. Lafferty, Statistical methods for text segmentation, *Machine Learning, Special Issue on Natural Language Learning*, **1-3** (1999).

[3] A. L. Berger, S. A. Della Pietra, V. J. Della Pietra, A maximum entropy approach to natural language processing, *Computational Linguistics*, **22**, 1 (1996), 39–72.

[4] E. Brill, Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging, *Computational Linguistics*, **21**, 4 (1995), 543–565.

[5] E. Brill, J. Wu, Jun, Classifier combination for improved lexical disambiguation, *Proc. of COLING-ACL'98*, Montreal, Canada, 1998, 191–195.

[6] W. Daelemans, J. Zavrel, P. Berck, S. Gillis, A memory-based part-of-speech tagger generator, *Proc. of 4th Workshop on Very Large Corpora*, Copenhagen, Denmark, 1996.

[7] T. Dietterich, Machine learning research: Four current directions, *AI Magazine*, Winter 1997, 97–136.

[8] T. Dietterich, *Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms*, http://www.cs.orst.edu/ tgd/cv/pubs.html.

[9] D. Elworthy, Tagset design and inflected languages, *Proc. of the ACL SIGDAT Workshop*, Dublin, Ireland (also available as cmp-lg archive 9504002).

[10] H. van Halteren, J. Zavrel, W. Daelemans, Improving data driven wordclass tagging by system combination, *Proc. of COLING-ACL'98*, Montreal, Canada, 1998, 491–497.

[11] Gh. Păun, *1994 sau Schimbarea care nu schimbă nimic*, Editura Ecce Homo, Bucureşti, 1993.

[12] A. Rathaparkhi, A maximum entropy part of speech tagger, *Proc. of EMNLP'96*, Philadelphia, Pennsylvania, 1996.

[13] D. Tufiş, Tiered tagging, *Research Report* no. 32, June, 1998, RACAI, Bucharest, 72pp (in Romanian).

[14] D. Tufiş, Tiered tagging and combined language models classifiers, in *Text, Speech and Dialogue* (F. Jelinek, E. Nöth, eds.), *Lecture Notes in Artificial Intelligence*, **1692**, Springer, 1999, 28–33.

[15] D. Tufiş, A.-M. Barbu, V. Pătraşcu, G. Rotariu, C. Popescu Camelia, Corpora and corpus-based morpho-lexical processing, in *Recent Advances in Romanian Language Technology* (D. Tufiş, P. Andersen, eds.), Editura Academiei, Bucureşti, 1997, 35–56 (also available at http://www.racai.ro/books)

[16] D. Tufiş, N. Ide, T. Erjavec, Standardized specifications, development and assessment of large morpho-lexical resources for six central and eastern European languages, *Proc. of 1st International Conference on Language Resources and Evaluation,* Granada, Spain, 1998, 233–240.

[17] D. Tufiş, O. Mason, Tagging Romanian texts: a case study for QTAG, a language independent probabilistic tagger, *Proc. of 1st International Conference on Language Resources and Evaluation,* Granada, Spain, 1998, 589–596.