From Word Alignment to Word Senses, via Multilingual Wordnets

Dan Tufiş

Abstract

Most of the successful commercial applications in language processing (text and/or speech) dispense with any explicit concern on semantics, with the usual motivations stemming from the computational high costs required for dealing with semantics, in case of large volumes of data. With recent advances in corpus linguistics and statistical-based methods in NLP, revealing useful semantic features of linguistic data is becoming cheaper and cheaper and the accuracy of this process is steadily improving. Lately, there seems to be a growing acceptance of the idea that multilingual lexical ontologies might be the key towards aligning different views on the semantic atomic units to be used in characterizing the general meaning of various and multilingual documents. Depending on the granularity at which semantic distinctions are necessary, the accuracy of the basic semantic processing (such as word sense disambiguation) can be very high with relatively low complexity computing. The paper substantiates this statement by presenting a statistical/based system for word alignment and word sense disambiguation in parallel corpora. We describe a word alignment platform which ensures text pre-processing (tokenization, POS-tagging, lemmatization, chunking, sentence and word alignment) as required by an accurate word sense disambiguation.

1 The Pervasive Ambiguity of Natural Languages

Most difficult problems in natural language processing stem from the inherent ambiguous nature of the human languages. Ambiguity is

©2006 by D. Tufiş

present at all levels of traditional structuring of a language system (phonology, morphology, lexicon, syntax, semantics) and not dealing with it at the proper level, exponentially increases the complexity of the problem solving. Currently, the state of the art taggers (combining various models, strategies and processing tiers) ensure no less than 97-98% accuracy in the process of morpho-lexical full disambiguation. For such taggers a 2-best tagging[1] is practically 100% correct.

One further step is the word sense disambiguation (WSD) process. In the traditional compositional semantics, the meaning of a complex expression is supposed to be derivable from the meanings of its parts, and the way in which those parts are combined. Depending on the representation formalisms for the word-meaning representation, various calculi may be considered for computing the meaning of a complex expression from the atomic representations of the word senses. Obviously, one should be able, before hand, to decide for each word in a text which of its possible meanings is, contextually, the right one.

Therefore, it is a generally accepted idea that the WSD task is highly instrumental (if not indispensable) in semantic processing of natural language documents.

It is almost a truism that more decision makers, working together, are likely to find a common solution superior to each solution individually found. Dieterich [1] discusses conditions under which different decisions (in his case classifications) may be combined for obtaining a better result. Essentially, a successful automatic combination method would require comparable performance on behalf of the decision makers and, additionally, that they would not make the similar errors. This idea has been exploited by various NLP researchers in language modelling, statistical POS tagging, parsing, word alignment, word sense disambiguation, etc.

The WSD problem can be stated as being able to associate to an ambiguous word ($\boldsymbol{w}$) in a text or discourse, the sense ($\boldsymbol{s}_k$) which is dis-

[1] In k-best tagging, instead of assigning each word exactly one tag (the most probable in the given context), it is allowed to have occasionally at most k-best tags attached to a word and if the correct tag is among the k-best tags, the annotation is considered to be correct.

tinguishable from other senses ($\boldsymbol{s}_1, \ldots, \boldsymbol{s}_{k-1}, \boldsymbol{s}_{k+1}, \ldots, \boldsymbol{s}_n$) prescribed for that word by a reference semantic lexicon. One such semantic lexicon (actually a lexical ontology) is Princeton WordNet [2] version 2.0[2] (henceforth PWN). PWN is a very fine-grained semantic lexicon currently containing 203,147 sense distinctions, clustered in 115,424 equivalence classes (synsets). Out of the 145,627 distinct words, 119,528 have only one single sense. However, the remaining 26,099 words are those that one would frequently meet in a regular text and their ambiguity ranges from two senses up to 36. Several authors considered that sense granularity in PWN is too fine-grained for the computer use, arguing that even for a human (native speaker of English) the sense differences of some words are very hard to be reliably (and systematically) distinguished. There are several attempts to group the senses of the words in PWN in coarser grained senses – *hyper-senses* – so that clear-cut distinction among them is always possible for humans and (especially) computers. We will refer in this paper to two hyper-sense inventories used in the BalkaNet project [3]. A comprehensive review of the WSD state-of the art at the end of 90's can be found in [4]. Stevenson and Wilks [5] review several WSD systems that combined various knowledge sources to improve the disambiguation accuracy and address the issue of different granularities of the sense inventories. SENSEVAL[3] series of evaluation competitions on WSD is a very good source on learning how WSD evolved in the last 6-7 years and where is it nowadays.

We describe a multilingual environment, containing several monolingual wordnets, aligned to PWN used as an interlingual index (ILI). The word-sense disambiguation method combines word alignment technologies, and interlingual equivalence relations in multilingual wordnets [6]. Irrespective of the languages in the multilingual documents, the words of interest are disambiguated by using the same sense-inventory labels. The aligned wordnets were constructed in the context of the European project BalkaNet. The consortium developed monolingual wordnets for five Balkan languages (Bulgarian, Greek, Romanian Serbian, and Turkish) and extended the Czech wordnet initially developed

[2] http://www.cogsci.princeton.edu/~wn/

[3] http://www.cs.unt.edu/~rada/senseval

in the EuroWordNet project [6]. The wordnets are aligned to PWN, taken as an interlingual index, following the principles established by the EuroWordNet consortium. The version of the PWN used as ILI is an enhanced XML version where each synset is mapped onto one or more SUMO [7] conceptual categories and is classified under one of the IRST domains [8]. In the present version of the BalkaNet ILI there are used 2066 SUMO distinct categories and 163 domain labels. Therefore, for our WSD experiments we had at our disposal three sense-inventories, with very different granularities: PWN senses, SUMO categories and IRST Domains.

2 Word Alignment

The word alignment is the first step (the hardest) in our approach for the identification of word senses. In order to reduce the search space and to filter out significant information noise, the context is reduced to the level of sentence. Therefore, a parallel text $< T_{L1}T_{L2} >$ is represented as a sequence of pairs of one or more sentences in language L1 (S^1_{L1} S^2_{L1}...S^k_{L1}) and one or more sentences in language L2 (S^1_{L2} $\mathrm{S}^2_{L2}\ldots\mathrm{S}^m_{L2}$) so that the two ordered sets of sentences represent reciprocal translations. Such a pair is called a translation alignment unit (or translation unit). The word alignment of a bitext is an explicit representation of the pairs of words $< w_{L1}w_{L2} >$ (called translation equivalence pairs) co-occurring in the same translation units and representing mutual translations. The general word alignment problem includes the cases where words in one part of the bitext are not translated in the other part (these are called *null alignments*) and the cases where multiple words in one part of the bitext are translated as one or more words in the other part (these are called expression alignments).

The input format is obtained from two raw texts that represent reciprocal translations. If not already sentence aligned, the two texts are aligned by a sentence aligner, similar to Moore's aligner [9] but which unlike it, is able to recover the non-one-to-one sentence alignments. The texts in each language are then tokenized, tagged and lemmatized. Frequently, the translation equivalents have the same part-of speech, but relying on such a restriction would seriously affect the alignment recall. However, when the translation equivalents have different parts of speech, this difference is not arbitrary. *POS affinities,* $\{p(POS_m^{L1}|POS_n^{L2}), p(POS_n^{L2}|POS_m^{L1})\}$, are easy to estimate and we use them to filter out improbable translation equivalents pairs.

The next pre-processing step is represented by the sentence chunking in both languages. The chunks are recognized by a set of regular expressions defined over the tagsets and they correspond to (non-recursive) noun phrases, adjectival phrases, prepositional phrases and verb complexes (analytical realization of tense, aspect mood and diathesis and phrasal verbs). The texts are further processed by a statistical dependency linking parser. Finally, the bitext is assembled as an XML document (XCES[4] compliant format), which is the standard input for most of our tools.

2.1 Two Aligners and Their Combination

We developed two quite different word aligners, motivated by two distinct objectives: the first one, called YAWA, was motivated by a project aiming at the development of an interlingually aligned set of wordnets while the other one was developed within an SMT ongoing project. The first one was used for validating, against a multilingual corpus, the interlingual synset equivalences and also for WSD experiments. Although, initially it was concerned only with open class words recorded in a wordnet, turning it into an "all words" aligner was not a difficult task. YAWA is a three stage lexical aligner that uses bilingual translation lexicons and phrase boundaries detection to align words of a given bitext. The translations lexicons are generated by a different module, TREQ [10], which generates translation equivalence hypotheses for the pairs of words (one for each language in the parallel corpus) which have been observed occurring in aligned sentences more than expected by chance. The hypotheses are filtered by a loglikelihood score threshold. Several heuristics (string similarity-cognates, POS affinities and

[4] http://www.cs.vassar.edu/XCES/

alignments locality[5]) are used in a competitive linking manner [11] to extract the most likely translation equivalents.

YAWA generates a bitext alignment by incrementally adding new links to those created at the end of the previous stage. The existing links act as contextual restrictors for the new added links. From one phase to the other, new links are added without deleting anything. This monotonic process requires a very high precision (at the price of a modest recall) for the first step. The next two steps are responsible for significantly improving the recall and ensuring an increased F-measure.

In the rest of this section we present in some details the various steps of the two aligners, evaluate them individually and finally describe the combination of the alignments produced by YAWA and MEBA and evaluate the result of the combination.

2.2 YAWA

2.2.1 YAWA Phase 1: Content Words Alignment

YAWA begins by taking into account only very probable links that will represent the skeleton alignment to be the input for the second phase. This alignment is done using outside resources such as translation lexicons and involves only the alignment of content words (nouns, verbs, adjective and adverbs).

The bitext to be word-aligned is concatenated to a reference parallel corpus containing the languages of concern. For Romanian-English we use an almost 1.5 million words parallel corpus. The concatenation of the target bitext to the reference corpus is required in case the target bitext is too small to provide reliable statistical evidence for the possible translation equivalents that are extracted by the TREQ module. The translation equivalence pairs are ranked according to an association score (i.e. log-likelihood, DICE, point-wise mutual information, etc.).

[5] The *alignments locality* heuristics exploits the observation made by several researchers that adjacent words of a text in the source language tend to align to adjacent words in the target language. A more strict alignment locality constraint requires that all alignment links starting from a chunk, in the one language end in a chunk in the other language.

We found that the best filtering of the translation equivalents was the one based on the log-likelihood (LL) score with a threshold of 9.

Each translation unit (pair of aligned sentences) of the parallel corpus is scanned for establishing the most likely links based on a competitive linking strategy that takes into account the LL association scores given by the TREQ translation lexicon. If a candidate pair of words is not found in the translation lexicon, we compute their orthographic similarity (cognate score [10]). If this score is above a predetermined threshold (we used the empirically found value of 0.43), the two words are treated as if they existed in the translation lexicon with a high association score (in practice we have multiplied the cognate score by 100 to yield association scores in the range 0..100).

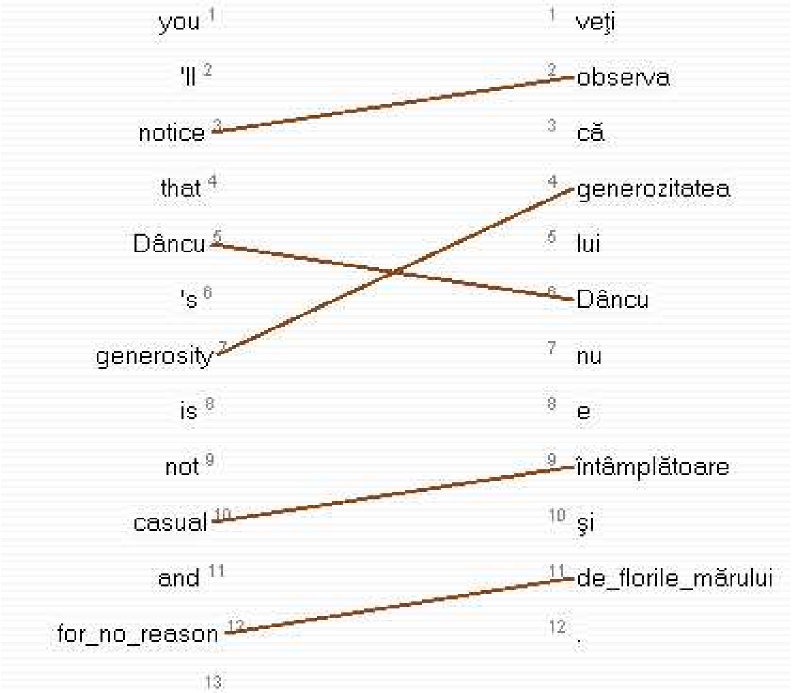


Figure 1. Alignment after the first step

The Figure 1 exemplifies the links created between two tokens of a parallel sentence by the end of the first phase.

2.2.2 YAWA Phase 2: Chunks Alignment

The second phase requires chunking of each part of the bitext. In our Romanian-English experiments, this requirement was fulfilled by using a set of regular expressions defined over the tagsets used in the target bitext. These simple chunkers recognize noun phrases, prepositional phrases, verbal and adjectival or adverbial groupings of both languages.

In this second phase YAWA firstly achieve the chunk-to-chunk matching and after that, continues with aligning the words of aligned chunks. Chunk alignment is done on the basis of the skeleton alignment produced in the first phase. The algorithm is simple: align two chunks $c(i)$ in source language and $c(j)$ in the target language if $c(i)$ and $c(j)$ have the same type (noun phrase, prepositional phrase, verbal group, adjectival/adverbial group) and if there exists a link $\langle w(s), w(t) \rangle$ so that $w(s) \in c(i)$ then $w(t) \in c(j)$.

After the chunks were aligned, a language pair dependent module takes over to align the unaligned words belonging to the chunks. Our module for the Romanian-English pair of languages contains some very simple empirical rules such as: if b is aligned to c and b is preceded by a, link a to c, unless there exist d in the same chunk with c and the POS category of d has a significant affinity with the category of a. The simplicity of these rules derives from the shallow structures of the chunks. In the above example b and c are content words while a is very likely a determiner or a modifier for b. The result of the second alignment phase, considering the same sentence from Figure 1, is exemplified in Figure 2. The new links are represented by the double lines:

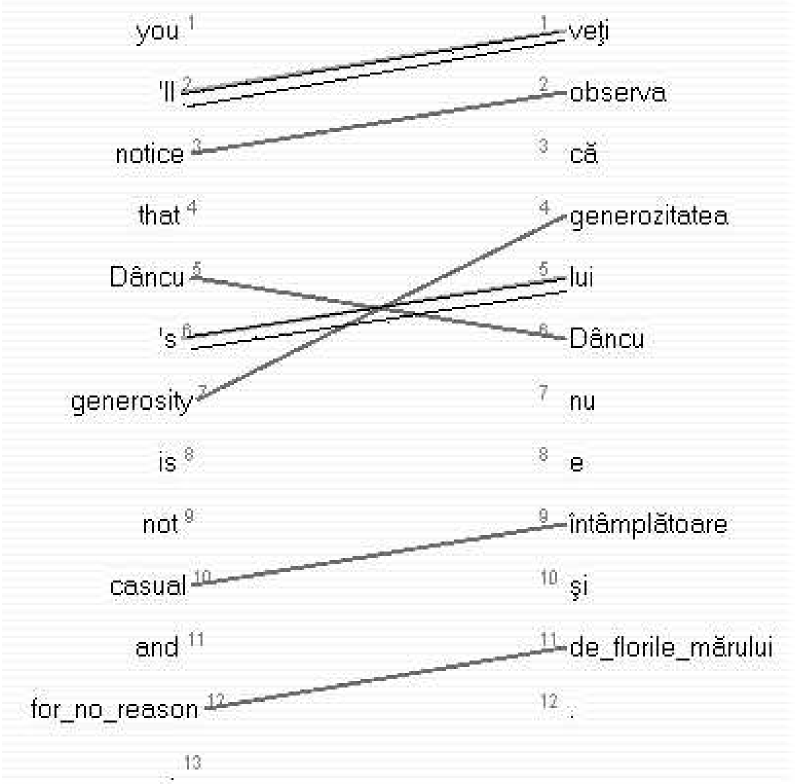


Figure 2. Alignment after the second step

2.2.3 YAWA Phase 3: Remaining Blocks of Words Alignment

This phase identifies contiguous sequences of words in each part of the bitext which remain unaligned and try to heuristically align the words of the best matching such blocks. The main criteria used is the POS-affinities of the remaining unaligned words and their relative positions. Let us exemplify, using the same example and the result shown in Figure 2, the way of adding new links in this last phase of the alignment. At the end of phase 2 the blocks of consecutive words that remain to be aligned are: English {en_1 = (that), en_2 = (is, not), en_3 = (and), en_4 = (.)} and Romanian {ro_1 = (că), ro_2 = (nu, e), ro_3 = (şi), ro_4 = (.)}. The mapping of source and target unaligned blocks depends on two criteria: the surrounding chunks are already aligned, and the pairs in the candidate unaligned blocks have significant POS-affinities. For instance in the Figure 2, blocks en_1 = (that) and ro_1 = (că) satisfy the above conditions because they appear among already aligned chunks (<'ll notice> ⇔ <veţi observa> and <Dâncu 's generosity> ⇔ <generozitatea lui Dâncu>) and they contain words with the same POS.

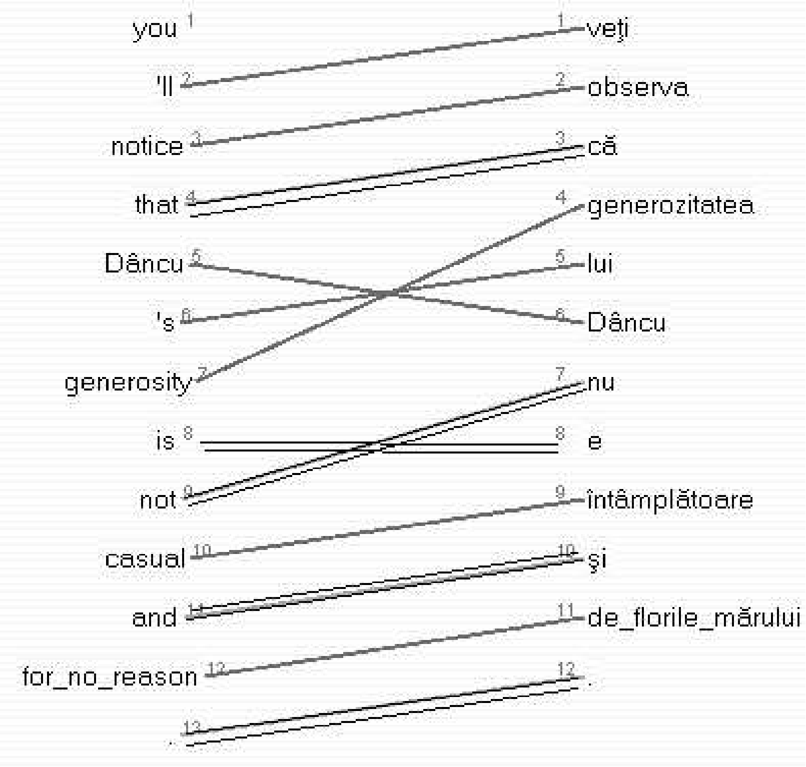


Figure 3. Alignment after the third step

After block alignment[6], given a pair of aligned blocks, the algorithm links words with the same POS and then the phase 2 is called again with these new links as the skeleton alignment. In Figure 3 is shown the result of phase 3 alignment of the sentence we used as an example throughout this section. The new links are shown (as before) by double lines.

The third phase is responsible for significant improvement of the alignment recall, but it also generates several wrong links. The detection of some of them is quite straightforward, and we added an additional correction phase 3.f. Romanian being a relatively free order language, it is quite easy to produce good quality translation by preserving the order of most of phrasal groups. We noticed this tendency in most of our training bilingual data so, we used this finding as an additional filter to remove those links that cross several regularly distributed links along the alignment.

2.2.4 YAWA Performance Analysis

The Table 1 presents the results of the YAWA aligner at the end of each alignment phase. Although the Precision decreases from one phase to the next one, the Recall gains are significantly higher so, the F-measure is monotonically increasing.

Table 1. YAWA evaluation

	Precision (P)	Recall (R)	F-Measure (F)
Phase 1	94.08%	34.99%	51.00%
Phase 1+2	89.90%	53.90%	67.40%
Phase 1+2+3	88.82%	73.44%	80.40%
Phase 1+2+3+3.f	**88.80%**	**74.83%**	**81.22%**

[6] Only 1:1 links are generated between blocks.

2.3 MEBA

2.3.1 MEBA Reifying Aligner

A quite different approach from the one used by YAWA, is implemented in our second word aligner, called **MEBA**. It is a multiple parameter and multiple step algorithm using relevance thresholds specific to each parameter, but different from each step to the other. The implementation of MEBA was strongly influenced by the famous five IBM models described in the [12] seminal paper. We used GIZA++ [13, 14] to estimate different parameters of the MEBA aligner.

MEBA is an iterative algorithm that takes advantage of all preprocessing phases mentioned in the beginning of the Section 2.

The alignment model considers a link between two candidate words as an object that is described by a feature-values structure (with values in the [0,1] interval) which we call the *reification* of the link. We differentiate between *context independent features* that refer only to the tokens of the current link (translation equivalency, part-of-speech affinity, cognates, etc.) and *context dependent features* that refer to the properties of the current link with respect to the rest of links in a bi-text (locality, number of traversed links, tokens indexes displacement, collocation). Also, we distinguish between bi-directional features (translation equivalency, part-of-speech affinity) and non-directional features (cognates, locality, number of traversed links, collocation, indexes displacement)

The program starts building the most probable links *(anchor links)*: cognates, numbers, dates, and translation pairs with high translation probabilities. Then, it iteratively aligns content words (open class categories) in the immediate vicinity of the anchor links. The links to be added at any later step are supported or restricted by the links created in the previous iterations. Each of the iterations can be configured to align different categories of tokens (named entities, dates and numbers, content words, functional words, punctuation) in decreasing order of statistical evidence, with different weights and different significance thresholds on each feature and iteration.

The score of a candidate link (LS) between a source token i and a target token j is computed by a linear function of the features scores:

$$LS(i,j) = \sum_{i=1}^{n} \lambda_i * ScoreFeat_i; \sum_{i=1}^{n} \lambda_i = 1$$

In the following subsection we briefly discuss the main features we use in reifying a link.

2.3.2 MEBA Features

In this section we will denote by A and B the source and target lexical items respectively.

Translation equivalence. The word aligner invokes GIZA++ to build translation probability lists for either lemmas or the word-forms of the bitext. The considered token for the translation model build by GIZA++ is the respective lexical item (lemma or word-form) trailed by its POS tag (eg. plane_N, plane_V plane_A). In this way we avoid data sparseness and filter noisy data. A further way of removing the noise created by GIZA++ is to filter out all the translation pairs below a LL-threshold. We made various experiments and empirically set the value of this threshold to 6. All the probability losses by this filtering were redistributed proportionally to their initial probabilities to the surviving translation equivalence candidates.

Translation equivalence entropy score. The translation equivalence entropy score is a favouring parameter for the words with a skewed probability distribution for their translation equivalents[7]. Since this feature is definitely sensitive to the order of the lexical items, we compute an average value for the link: αES(A)+βES(B). Currently we use $\alpha = \beta = 0.5$, but it might be interesting to see, depending on different language pairs, how the performance of the aligner would be affected by different settings of these parameters.

$$ES(W) = 1 - \frac{-\sum_{i=1}^{N} p(W,TR_i) * \log p(W,TR_i)}{\log N}$$

[7] This heuristics implements the zipffian conjecture about the word senses distribution in a coherent text.

Part-of-speech affinity. In faithful translations the translated words tend to be translated by words of the same part-of-speech. When this is not the case, the different POSes, are not arbitrary. The part of speech affinity, P(POS(A)|POS(B)), can be easily computed from a gold standard alignment. Obviously, this is a directional feature, so an interpolation operation is necessary in order to ascribe this feature to a link:

$$\text{PA}{=}\alpha\ \text{P(POS(A)|POS(B))} + \beta\ \text{P(POS(A)|POS(B))}.$$

Again, we used $\alpha = \beta = 0.5$ but different values of these weights might be worthwhile investigating.

Cognates. The similarity measure, COGN(A, B), is implemented as a Levenstein metric. Using the COGN test as a filtering device is a heuristic based on the *cognate conjecture* which says that when the two tokens of a translation pair are orthographically similar, they are very likely to have similar meanings (i.e. they are cognates). The threshold for the COGN(A, B) test was empirically set to 0.43. This value depends on the pair of languages in the bitext. The actual implementation of the COGN test includes a language-dependent normalisation step, which strips some suffixes, discards the diacritics, reduces some consonant doubling, etc. This normalisation step was hand written, but, based on available lists of cognates, it could be automatically induced.

Obliqueness. Each token in both sides of a bitext is characterized by a position index, computed as the ratio between the relative position in the sentence and the length of the sentence. The absolute value of the difference between tokens' position indexes, subtracted from 1 gives the link's "obliqueness".

$$OBL(A_i, B_j) = 1 - \left| \frac{i}{length(Sent_S)} - \frac{j}{length(Sent_T)} \right|$$

Locality is a feature that estimates the degree to which the links are sticking together.

MEBA has three features to account for locality: (i) *weak locality*, (ii) *chunk-based locality* and (iii) *dependency-based* locality (see Figure 4).

The value of the *weak locality* feature is derived from the already existing alignments in a window of N tokens centred on the focused token. The window size is variable, proportional to the sentence length. If in the window there exist k linked tokens and the indexes of their links are $< i_1 j_1 >, \ldots < i_k j_k >$ then the locality feature of the new link $< i_{k+1}, j_{k+1} >$ is defined by the equation below:

$$LOC = \min(1, \frac{1}{k}\sum_{m=1}^{k}\frac{|i_{k+1} - i_m|}{|j_{k+1} - j_m|}).$$

In the case of *chunk-based locality* the window span is given by the indexes of the first and last tokens of the chunk.

Dependency-based locality uses the set of the dependency links of the tokens in a candidate link for the computation of the feature value. In this case, the LOC feature of a candidate link $< i_{k+1}, j_{k+1} >$ is set to 1 or 0 according to the following rule:

> if between i_{k+1} and i_α there is a (source language) dependency and if between j_{k+1} and j_β there is also a (target language) dependency then LOC is 1 if i_α and j_β are aligned, and 0 otherwise.
>
> Note that in case $j_{k+1} \equiv j_\beta$ a trivial dependency (identity) is considered and the LOC attribute of the link $< i_{k+1}, j_{k+1} >$ is set always to 1.

Collocation. We used the bi-grams list to annotate the chains of lexical dependencies among the contents words. Then, the value of the collocation feature is computed similar to the dependency-based locality feature. The algorithm searches for the links of the lexical dependencies around the candidate link.

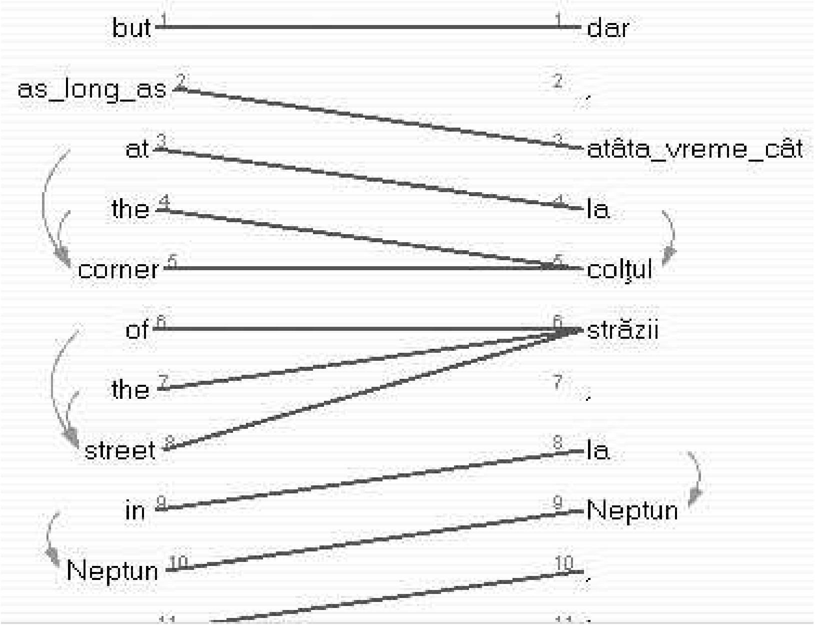


Figure 4. Chunk and dependency-based locality

Monolingual collocation is an important clue for word alignment. If a source collocation is translated by a multiword sequence, very often the lexical cohesion of source words can also be found in the corresponding translated words. In this case the aligner has strong evidence for many to many linking. When a source collocation is translated as a single word, this feature is a strong indication for a many to 1 linking.

Bi-gram lists (only content words) were built from each monolingual part of the training corpus, using the log-likelihood score (threshold of 10) and minimal occurrence frequency (3) for candidates filtering.

We used the bi-grams list to annotate the chains of lexical dependencies among the contents words (see Figure 5). Then, the value of the collocation feature is computed similar to the dependency-based locality feature. The algorithm searches for the links of the lexical dependencies around the candidate link.

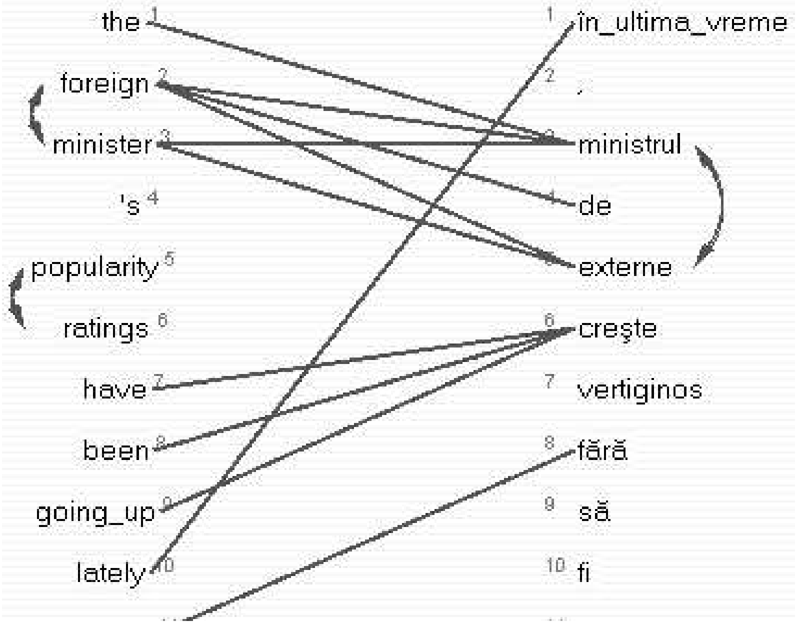


Figure 5. Collocation feature

2.3.3 MEBA Performance Analysis

The cumulative results of the major processing steps are shown in the table below. As one can see the precision decreases from the first step to the last with 6.25% but the recall (almost three times better) and the F-measure (almost double) are significantly improved.

The alignments generated by MEBA were compared to the ones produced by YAWA and evaluated against the Gold Standard (GS) annotations used in the Word Alignment Shared Tasks (Romanian-English track) organized at HLT-NAACL2003 [15].

As one can observe from the results in Table 1 and Table 2 the two aligners, which are based on quite different models, have comparable performances. Moreover, by analyzing the alignment errors done by

Table 2. MEBA evaluation

	Precision	Recall	F-measure
“Anchor” links	98.40%	28.82%	44.58%
Words around “anchors”	96.28%	44.32%	60.70%
Functional words and punctuation	93.23%	61.98%	74.46%
Probable links	**92.15%**	**73.40%**	**81.71%**

each word aligner, we found that the number of common mistake was small so, the premises for a successful combination were very good [1].

2.4 COWAL: The Combined Aligner

The Combined Word Aligner, **COWAL**, is a wrapper of the two aligners (YAWA and MEBA) merging the individual alignments and filtering the result. At the Shared Task on Word Alignment organized by the ACL2005 Workshop on “Building and Using Parallel Corpora: Data-driven Machine Translation and Beyond” [16], we participated (on the Romanian-English track) with the two aligners and the combined one (COWAL). Out of 37 competing systems, COWAL [17] was rated the first, MEBA the 20^{th} and TREQ-AL [18], (the former version of YAWA), was rated the 21^{st}. The usefulness of the aligner combination was convincingly demonstrated. Meanwhile, both the individual aligners were significantly improved as well as their combination.

One very simple, but very effective method of alignment combination is a heuristic procedure which merges the alignments produced by two or more word aligners and filters out the links that are likely to be wrong. For the purpose of filtering, a link is characterized by its type defined by the pair of indexes (i,j) and the POS of the tokens of the respective link. The likelihood of a link is proportional to the POS affinities of the tokens of the link and inverse proportional to the *bounded relative positions* (BRP) of the respective tokens: $BRP = 1 + ||i - j| - avg|$ where avg is the average displacement in a Gold Standard of the aligned tokens with the same POSes as the tokens of the current link. From the same gold standard we estimated a threshold below which a link is removed from the final alignment.

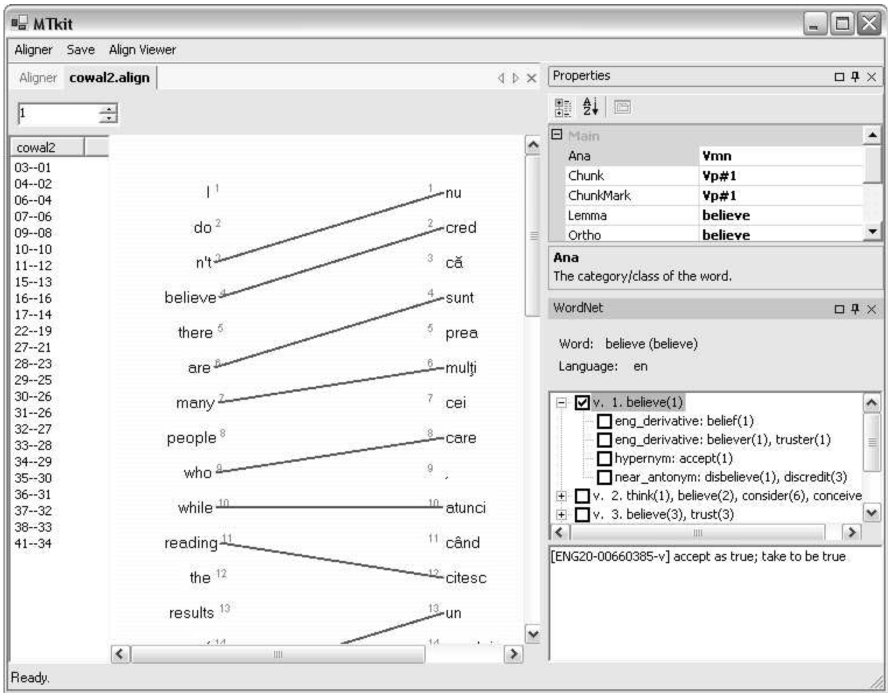
A more elaborated alignment combination (with better results than the previous one) is modelled as a binary statistical classification problem (good / bad) and, as in the case of the previous method, the net result is the removal of the links which are likely to be wrong. We used the SVM training and classification toolkit - LIBSVM [19] with the default parameters (C-SVC classification and radial basis kernel function). Both context independent and context dependent features characterizing the links were used for training. The classifier was trained with positive and negative examples of links. A subset of the Gold Standard alignment links was used as positive examples set. The same number of negative examples was extracted from the alignments produced by COWAL and MEBA where they differ from the Gold Standard.

The result of the SVM-based combination (COWAL), compared with the individual aligners, is shown in Table 3.

Table 3. Combined alignment

Aligner	P	R	F-measure
YAWA	88.80%	74.83%	81.22%
MEBA	92.15%	73.40%	81.71%
COWAL	87.26%	80.94%	83.98%

COWAL is now embedded into a larger platform (called MTkit) that incorporates the tools for bitexts pre-processing, a graphical interface that allows for comparing and editing different alignments, as well as a word sense disambiguation module. A snapshot of the COWAL graphical interface is shown in Figure 6. The left pane in Figure 6 is the alignment viewer and editor area. The user can edit the alignments (delete and add one or multiple links). By double clicking a word in this pane, its properties will be automatically displayed in the right-hand windows. The upper-right window shows the lexico-syntactic properties of the selected word: the morphological analysis of the orthographic form, its lemma, the syntactic chunk to which it belongs. Currently this pane is not editable. The bottom-right window displays the semantic properties of the selected word: its sense in the current context, the gloss for this sense, synonyms, hyperonyms, derivatives, etc. These properties are extracted from the wordnet of the language to which the selected word belongs to. This pane is editable, but only the sense number is subject to user modifications.



Ana	Vmn
Chunk	Vp#1
ChunkMark	Vp#1
Lemma	believe
Ortho	believe

Ana
The category/class of the word.

Word: believe (believe)
Language: en

- ☑ v. 1. believe(1)
  - □ eng_derivative: belief(1)
  - □ eng_derivative: believer(1), truster(1)
  - □ hypernym: accept(1)
  - □ near_antonym: disbelieve(1), discredit(3)
- □ v. 2. think(1), believe(2), consider(6), conceive
- □ v. 3. believe(3), trust(3)

[ENG20-00660385-v] accept as true; take to be true

Figure 6. COWAL Graphical User Interface

Although far from being perfect, the accuracy of word alignments and of the translation lexicons extracted from parallel corpora is rapidly improving. In the shared task evaluations of different word aligners, organized on the occasion of the 2003 NAACL Conference and the 2005 ACL Conference, our winning systems TREQ-AL [18] and COWAL [17] produced wordnet-relevant lexicons[8] with F-measures as high as 84.26% and 89.92%[9].

3 Wordnet-based Sense Disambiguation

The task of word sense disambiguation (WSD) requires one reference sense inventory in terms of which the senses of the target words will be labeled. We argued at length elsewhere [20] that a meaningful discussion of the performances of a WSD system cannot dispense of clearly specifying the sense inventory it uses, and the comparison between two WSD systems that uses different sense inventories is frequently more confusing than illuminating. Essentially, this is because the differences in the semantic distinctions (sense granularities), as used by different semantic dictionaries (sense repositories), make the difficulty of the WSD task range over a large spectrum. For instance, the discrimination of homographs (more often than not having different parts of speech, e.g. "(to) bottle" as storing liquids or gases in bottles, versus "bottle" as the recipient) is much simpler than metonymic distinctions (e.g. "bottle" as container, versus "bottle" as content).

In our research, we used the Princeton Wordnet 2.0 as the major sense inventory and the BalkaNet multilingual lexical ontology. The BalkaNet lexical ontology has been developed within the European project with the same name (September 2001-August 2004) and includes five languages from the Balkan area (Bulgarian, Greek, Romanian, Serbian and Turkish), plus Czech, the wordnet of which, initially developed in EuroWordNet, has been significantly extended. By observing the interlingual synset mapping principle and incorporating most of the conceptual extensions proposed by EuroWordNet, the

[8] wordnet-relevant lexicons are restricted only to translation pairs of the same major POS (nouns, verbs, adjectives and adverbs).

[9] Currently, with the most recent improvements, COWAL's F-measure is 92.08%

BalkaNet wordnets can be easily combined with any of the other semantic networks of the EuroWordnet, and, thus, one may speak about a really pan-European multilingual lexical ontology, covering at least 15 languages[10].

The BalkaNet multilingual environment took advantage of the latest developments in the PWN that was adopted itself as an interlingual index. This is a major difference with respect to the EuroWordNet's ILI. As the SUMO/MILO [7] and DOMAINS [8], have been aligned with PWN, they automatically became available in each monolingual wordnet of the BalkaNet. To allow the representation of language idiosyncratic properties, structural knowledge present in the monolingual wordnets has precedence over the structural knowledge imported from the ILI. As the Romanian wordnet imported SUMO/MILO and DOMAINS labels and the synsets unique identifiers are the same as in the PWN, it is self-contained but at the same time unambiguously integratable in a PWN centered multilingual wordnet infrastructure.

Once the translation equivalents identified, it is reasonable to expect that the words of a translation pair $< w^i_{L1}, w^j_{L2} >$ share at least one conceptual meaning stored in an interlingual sense inventory. When interlingually aligned wordnets are available (as is our case), obtaining the sense labels for the words in a translation pair is straightforward: one has to identify for w^i_{L1} the synset S^i_{L1} and for w^j_{L2} the synset S^j_{L2} so that S^i_{L1} and S^j_{L2} are projected over the same interlingual concept. The index of this common interlingual concept (ILI) is the sense label of the two words w^i_{L1} and w^j_{L2}. However, it is possible that no common interlingual projection will be found for the synsets to which w^i_{L1} and w^j_{L2} belong. In this case, the senses of the two words will be given by the indexes of the most similar interlingual concepts corresponding to the synsets of the two words. Our measure of interlingual concepts semantic similarity is based on PWN structure. We compute the semantic-similarity[11] score by the formula $SYM(ILI_1, ILI_2) = \frac{1}{1+k}$

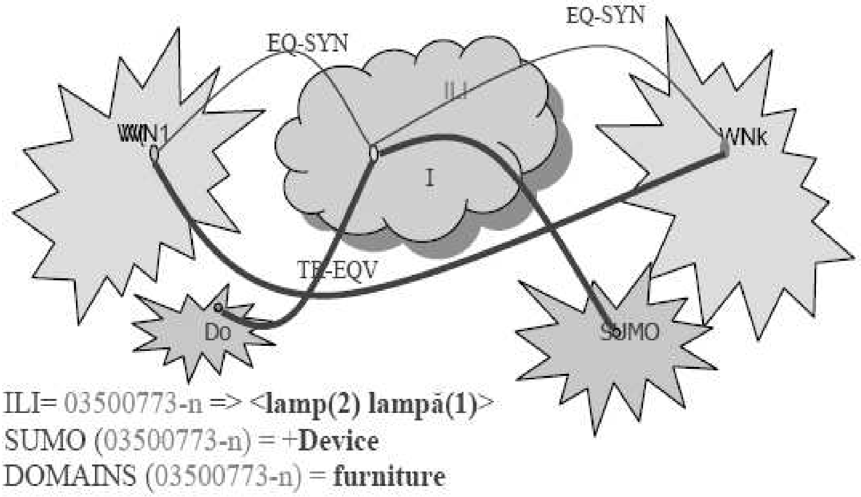
[10] Basque, Bulgarian, Catalan, Dutch, Czech, English, Estonian, French, German, Greek, Italian, Romanian, Serbian, Spanish, and Turkish.

[11] For a detailed discussion and an in-depth analysis of several other measures see: Budanitsky, A., Hirst, G., Semantic distance in WordNet: An experimental,

where k is the number of links from ILI_1 to ILI_2 or from both ILI_1 and ILI_2 to the nearest common ancestor. In Figure 7 and Figure 8, we exemplify the process of sense labeling of the words in two translation pairs as detected by the word alignment phase.

Let us consider first the pair <**lamp, lampă**>. Looking up the English and Romanian wordnets for the synsets that contain the words “lamp” and “lampă” respectively, we find the following lists of unique identifiers that differentiate among the noun senses of the two words:

PWN2.0 (lamp) = {03500372-n, 03500773-n}
RoWN (lampă) = {03500773-n, 03500872-n}



ILI= 03500773-n => <**lamp(2) lampă(1)**>
SUMO (03500773-n) = +**Device**
DOMAINS (03500773-n) = **furniture**

Figure 7. <**lamp lampă**>

The intersection reveals one common identifier (03500773-n) which, therefore, is taken as the common interlingual meaning. From an ILI

application-oriented evaluation of five measures. Proceedings of the Workshop on WordNet and Other Lexical Resources, NAACL, Pittsburgh, June, (2001) 29-34.

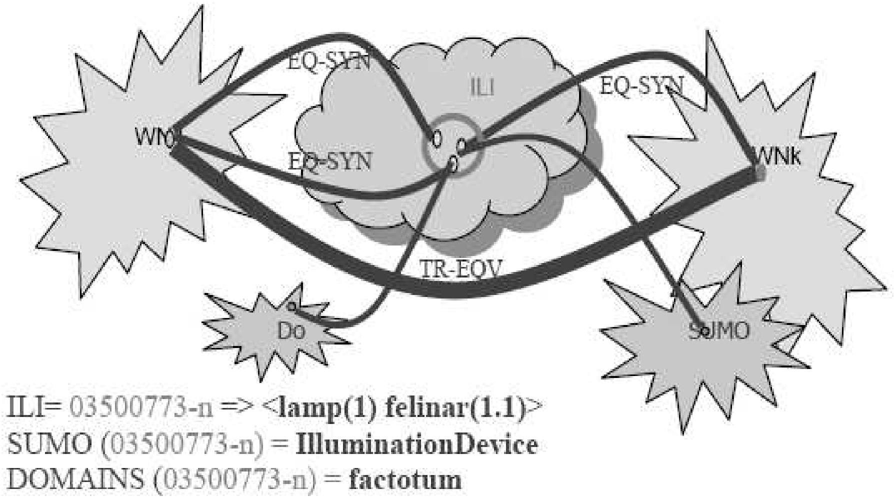
code, one can deterministically determine the SUMO concept and the DOMAINS label (see Figure 7).

Now, if we consider a different translation equivalent for the word **"lamp"**, namely **"felinar"** and repeat the procedure described above,

PWN2.0 (lamp) = {03500372-n, 03500773-n}

RoWN (felinar) = {003505057-n}

we notice that there is no common interlingual ILI code in the two lists. In this case, the metrics mentioned above is used to select the closest related senses: SYM(03500372-n,003505057-n)=0.5; SYM(03500373-n,003505057-n)=0.125 (see Figure 8).



ILI= 03500773-n => <**lamp(1) felinar(1.1)**>
SUMO (03500773-n) = **IlluminationDevice**
DOMAINS (03500773-n) = **factotum**

Figure 8. <**lamp felinar**>

After the WSD process has finished, the sense information is inserted into the XML encoding of the corpus. Which sense inventory (ILI, SUMO or DOMAINS) should be used in the encoding is a user-set parameter, which by default includes all of them.

In Figure 9, it is shown the final encoding of one translation unit of the “1984” parallel corpus. The “sn” attribute represents the Princeton Wordnet 2.0 unique synset identifier (ILI code), the “oc” attribute represents the SUMO ontology concept and the “dom” attribute represents the DOMAINS label.

```
<tu id="Ozz20">
 <seg lang="en">
 <s id="Oen.1.1.4.9">
  <w lemma="the" ana="Dd">The</w>
  <w lemma="patrol" ana="Ncnp" sn=“3" oc=“Group" dom="military">patrols</w>
  <w lemma="do" ana="Vais">did</w>
  <w lemma="not" ana="Rmp" sn="1" oc="not“ dom="factotum">not</w>
  <w lemma="matter" ana="Vmn" sn="1" oc="SubjAssesAttr" dom="factotum">matter</w>
  <c>,</c>
  <w lemma="however" ana="Rmp" sn="1" oc="SubjAssesAttr|PastFn” dom="factotum">however</w>
  <c>.</c>
 </s>
</seg>
<seg lang="ro">
 <s id="Oro.1.2.5.9">
  <w lemma="şi" ana=Crssp>Şi</w>
  <w lemma="totuşi" ana="Rgp" sn="1“ oc="SubjAssesAttr|PastFn" dom="factotum">totuşi</w>
<c>,</c>
  <w lemma="patrulă” ana="Ncfpry" sn="1.1.x" oc=“Group" dom="military"> patrulele</w>
  <w lemma="nu" ana="Qz" sn="1.x" oc="not" dom="factotum">nu</w>
  <w lemma="conta" ana="Vmii3p" sn="2.x" oc="SubjAssesAttr" dom="factotum"> contau</w>
  <c>.</c>
 </s>
</seg>
...
    </tu>
```

Figure 9. The final corpus encoding

4 WSD Evaluation

The BalkaNet version of the “1984” corpus is encoded as a sequence of uniquely identified *translation units*. For the evaluation purposes, we selected a set of frequent English words (123 nouns and 88 verbs) the meanings of which were also encoded in the Romanian wordnet. The selection considered only polysemous words (at least two senses per part of speech) since the POS-ambiguous words are irrelevant as this distinction is solved with high accuracy (more than 99%) by our tiered-tagger [21]. All the occurrences of the target words were disambiguated by three independent experts who negotiated the disagreements and thus created a gold-standard annotation for the evaluation of precision and recall of the WSD algorithm. The Table 4 summarizes the results.

Table 4. WSD precision, recall and F-measure

Precision	**Recall**	**F-measure**
78.21%	78.21%	78.21%

With the PWN senses identified (synset unique identifiers), sense labeling with either SUMO and/or IRST domains inventories is trivial, as described before, because the synset unique identifiers of PWN are already mapped (clustered) onto these two sense inventories. The Table 5 shows a great variation in terms of Precision, Recall and F-measure when different granularity sense inventories are considered for the WSD problem. Thus, it is important to make the right choice on the sense inventory to be used with respect to a given application. In case of a document classification problem, it is very likely that the IRST domain labels (or a similar granularity sense inventory) would suffice. The rationale is that IRST domains are directly derived from the Universal Decimal Classification as used by most libraries and librarians. The SUMO sense labeling will be definitely more useful in an ontology based intelligent system interacting through a natural language interface. Finally, the most refined sense inventory of PWN will be extremely useful in Natural Language Understanding Systems, which would require a deep processing. Such a fine inventory would be highly beneficial in lexicographic and lexicological studies.

Table 5. Evaluation of the WSD in terms of three different sense inventories.

Sense Inventory	**Precision**	**Recall**	**F-measure**
PWN 115424 categories	78.21%	78.21%	78.21%
SUMO 2066 categories	85.08%	85.08%	85.08%
DOMAINS 163 categories	93.30%	93.30%	93.30%

Similar findings on sense granularity for the WSD task are discussed in [5] where for some coarser grained inventories even higher precisions are reported. However, we are not aware of better results in WSD exercises where the PWN sense inventory was used. The major explanation for this is that unlike the majority work in WSD that is based on monolingual environments, we use for the definition of sense contexts the cross-lingual translations of the occurrences of the target words. The way one word in context is translated into one or more other languages is a very accurate and highly discriminative knowledge source for the decision-making.

5 Conclusions

Word Alignment is a highly promising technology with real prospects of soon reaching full maturity and reliability as needed by commercial applications. Among them, one could mention multilingual computational lexicography and terminology, multilingual documents indexing and retrieval, open domain natural language question answering and obviously machine translation. We described another application, WSD, which is not an end in itself, but necessary at one level or another to accomplish most natural language processing tasks.

Neither YAWA nor MEBA needs an a priori bilingual dictionary, as this will be automatically extracted by the TREQ or GIZA++. We made evaluation of the individual alignments in both experimental settings: without a startup bilingual lexicon and with an initial mid-sized bilingual lexicon. Surprisingly enough, we found that while the performance of YAWA increases a little bit (approx. 1% increase of the F-measure) MEBA is doing better without an additional lexicon. So, in the evaluation presented in the previous section MEBA uses only the training data vocabulary. The automatically extracted lexicons, could be almost 100% accurate (with a sufficiently high occurrence threshold) which is obviously a very good starting point in compiling bilingual dictionaries for language pairs where such electronic resources are not easily available.

YAWA is very sensitive to the quality of the bilingual lexicons it uses. We used automatically translation lexicons (with or without a seed lexicon), and the noise inherently present might have had a bad influence on YAWA's precision. Replacing the TREQ-generated bilingual lexicons with validated (reference bilingual lexicons) would further improve the overall performance of this aligner. Yet, this might be a harder to meet condition for some pairs of languages than using parallel corpora.

MEBA is more versatile as it does not require a-priori bilingual lexicons but, on the other hand, it is very sensitive to the values of the parameters which control its behaviour. Currently they are set according to the developers' intuition and after the analysis of the results from several trials. Since this activity is pretty time consuming (human analysis plus re-training might take a couple of hours) we plan to extend MEBA with a supervised learning module, which would automatically determine the "optimal" parameters (thresholds and weights) values.

The results in Table 5 show that although we used the same WSD algorithm on the same text, the performance scores (precision, recall, f-measure) significantly varied, with more than 15% difference between the best (DOMAINS) and the worst (PWN) f-measures. This is not surprising, but it shows that it is extremely difficult to objectively compare and rate WSD systems working with different sense inventories.

The potential drawback of this approach is that it relies on the existence of parallel data and at least two aligned wordnets that might not be available yet. Nevertheless, parallel resources are becoming increasingly available, in particular on the World Wide Web, and aligned wordnets are being produced for more and more languages (currently there are more than 40 ongoing wordnet projects for 37 languages). In the near future it should be possible to apply our and similar methods to large amounts of parallel data and a wide spectrum of languages.

Acknowledgements

The reported work is the result of several year intensive research at our institute. Many people deserve acknowledgements here, but special mentioning is due to Radu Ion, Alin Ceauşu, Dan Ştefănescu, Verginica Barbu-Mititelu and Elena Irimia, currently preparing their PhD theses on topics directly or closely related to those discussed in this paper.

References

[1] Dieterich, T., *G. Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms,* Neural Computation, vol. 10, no. 7, pp. 1895–1923, 1998

[2] Fellbaum, Ch. (ed.) *WordNet: An Electronic Lexical Database,* MIT Press (1998).

[3] Tufiş, D. (ed): *Special Issue on BalkaNet.* Romanian Journal on Science and Technology of Information, Vol. 7 no. 3-4 (2004) 9–44.

[4] Ide, N., Veronis, J., *Introduction to the special issue on word sense disambiguation. The state of the art.* Computational Linguistics, Vol. 27, no. 3, (2001) 1–40.

[5] Stevenson, M., Wilks, Y., *The interaction of Knowledge Sources in Word Sense Disambiguation.* Computational Linguistics, Vol. 24, no. 1, (1998) 321–350.

[6] Vossen P. (ed.) *A Multilingual Database with Lexical Semantic Networks, Kluwer Academic Publishers,* Dordrecht, 1998

[7] Niles, I., and Pease, A., *Towards a Standard Upper Ontology.* In Proceedings of the 2^{nd} International Conference on Formal Ontology in Information Systems (FOIS-2001), Ogunquit, Maine, (2001) 17–19.

[8] Magnini B. Cavaglià G., *Integrating Subject Field Codes into WordNet.* In Proceedings of LREC2000, Athens, Greece (2000) 1413–1418.

[9] Moore, R. 2002. *Fast and Accurate Sentence Alignment of Bilingual Corpora in Machine Translation: From Research to Real Users.* In Proceedings of the 5th Conference of the Association for Machine Translation in the Americas, Tiburon, California), Springer-Verlag, Heidelberg, Germany: 135–244.

[10] Tufiş, D. *A cheap and fast way to build useful translation lexicons.* In Proceedings of the 19th International Conference on Computational Linguistics, COLING2002, Taipei, 25-30 August, 2002, pp. 1030–1036, ISBN 1-55860-894.

[11] Melamed, D. *Empirical Methods for Exploiting Parallel Texts.* Cambridge, MA: MIT Press, 2001

[12] Brown, P. F., Della Pietra, S.A., Della Pietra, V. J., Mercer, R. L.(1993) *The mathematics of statistical machine translation: Parameter estimation.* Computational Linguistics, 19(2) pp. 263–311

[13] Och, F., J., Ney, H., *Improved Statistical Alignment Models,* Proceedings of ACL2000, Hong Kong, China, 440–447, 2000.

[14] Och, F.J., Ney, H. *A Systematic Comparison of Various Statistical Alignment Models,* Computational Linguistics, 29(1), pp. 19–51, 2003

[15] Rada Mihalcea and Ted Pedersen, *An Evaluation Exercise for Word Alignment,* in Proceedings of the HLT/NAACL Workshop

on Building and Using Parallel Texts: Data Driven Machine Translation and Beyond, Edmonton, Canada, May 2003.

[16] Martin, J., Mihalcea, R., Pedersen, T. *Word Alignment for Languages with Scarce Resources.* In Proceeding of the ACL2005 Workshop on "Building and Using Parallel Corpora: Data-driven Machine Translation and Beyond". June, 2005, *Ann Arbor, Michigan, June,* Association for Computational Linguistics, 65–74.

[17] Tufiş, D., Ion, R. Ceauşu, Al., Stefănescu, D.: *Combined Aligners.* In Proceeding of the ACL2005 Workshop on "Building and Using Parallel Corpora: Data-driven Machine Translation and Beyond". June, 2005, *Ann Arbor, Michigan, June,* Association for Computational Linguistics, pp. 107–110.

[18] Tufiş, D., Barbu, A., M., Ion, R. *A word-alignment system with limited language resources.* In Proceedings of the NAACL 2003 Workshop on Building and Using Parallel Texts; Romanian-English Shared Task, Edmonton (2003) 36–39.

[19] Fan, R., Chen, P.H, Lin, C.J. *Working set selection using the second order information for training SVM. Technical report 2005.,* Department of Computer Science, National Taiwan University (www.csie.ntu.edu.tw/ ~cjlin/papers/quadworkset.pdf).

[20] Tufiş, D., Ion, R. *Evaluating the word sense disambiguation accuracy with three different sense inventories.* In Proceedings of the Natural Language Understanding and Cognitive Systems Symposium, Miami, Florida, May 2005, pp. 118–127, ISBN 972-8865-23-6

[21] Tufiş, D., *Tiered Tagging and Combined Classifiers, in F. Jelinek, E. Nöth (eds) Text, Speech and Dialogue,* Lecture Notes in Artificial Intelligence, Vol. 1692. Springer-Verlag, Berlin Heidelberg New-York (1999) 28–33.

Dan Tufiş,

Received January 5, 2006

Institute for Artificial Intelligence,
13, "13 Septembrie", 050711, Bucharest 5, Romania
E–mail: *tufis@racai.ro*