Syntactic Disambiguation by Learning Weighted Government Patterns from a Large Corpus

Alexander F. Gelbukh [1]
Igor A. Bolshakov [2]
Sofía N. Galicia-Haro [3]

[1, 3] Natural Language Processing Laboratory,
Center for Computer Research,
National Polytechnic Institute,
Mexico City.
{gelbukh, sofia}@pollux.cic.ipn.mx

[2] Computer fund of Russian language,
Institute for Russian Language,
Russian Academy of Sciences,
Moscow, Russia.
iabolsh@aha.ru, igor@pollux.cic.ipn.mx

Abstract

A method of syntactic disambiguation based on proper prepositional phrase attachment, or, more generally, attachment of the clauses in specific grammatical cases, is described. The research was based on Spanish and Russian material. The data set built and used by the procedure is a kind of a syntactic government patterns dictionary. The algorithm requires a morphological and a syntactic parser and assigns probability weights to the variants built by the parser. No manual markup is required. At the training stage, the procedure works iteratively on a large text corpus, in alternating steps re-estimating the frequencies of individual government patterns and then the weights of the variants. The method is compatible with other methods of estimation of the variants. The data set built by the algorithm is useful for compilation of a combinatory dictionary for human readers. Some generalizations of the method are discussed.

1. Introduction*

Syntactic ambiguity is one of the most difficult problems of text processing and processing of large text corpora. In many languages syntactic ambiguity is greatly increased by ambiguity of attachment of prepositional phrases, or, more generally, of clauses in specific grammatical cases. Our research was based mainly on Spanish and Russian material, but all the results are fully applicable to English, with the exception that we do not consider disambiguation of the attributive chains, which present an additional source of syntactic ambiguity specific for English. To keep the size of the article reasonable, other simplifications are made: (1) the passive transformations, as well as Spanish and Russian impersonal and reflexive constructions, are not discussed, (2) the handling of morphological ambiguity is not described.

* Parts of this work were done in frame of the project *Syntactical analyzer for Spanish language, driven by a dictionary of management patterns* supported by CONACyT, Mexico.

We show that a special data set of lexical nature is useful to resolve the ambiguity related to the use of prepositions and grammatical cases. The same data set, namely a kind of a combinatorial dictionary, is also necessary for text generation and even is very useful for foreigners learning the language or composing texts in it. We propose an iterative procedure to automatically learn such a data set from a large text corpus and simultaneously resolve the syntactic ambiguity in this corpus. The data set can than be used for disambiguation of other texts; we also have used it as a raw material for compilation of a human-oriented dictionary of government patterns.

There was significant interest in the last years to the problem of prepositional phrase attachment, mainly on English material, involving both rule-based approach [6] and statistical lexical approaches [8, 12]. On the other hand, various iterative and re-estimation methods were used to calculate the probabilities used in hidden Markov models, such as Baum-Welch re-estimation method [2], or to learn the grammar information from a corpus [13].

While these works are based mainly on "tagging" approach to parsing, in our paper we investigate the problem from the point of view of general problem of syntactic disambiguation, i.e., of choosing one of the possible syntactic trees for the whole phrase. We also connect the technical task of disambiguation with well-known linguistic notion of government patterns, and show that the data obtained in disambiguation of a large corpus can be used in semi-automatic compilation of a kind of a combinatory dictionary. We also introduce the idea of taking into account the probabilities of the typical errors made by the parser, in addition to the probabilities of some natural language constructions.

2. Ambiguity of prepositional phrase attachment

Let us consider a simple English phrase: *They moved their office from the town to the capital*. The morphological representation of the phrase usually operated upon by the parsers is: *NG V NG p NG p NG*, where *NG* is a noun group or a pronoun, *V* is a verb, *p* is a preposition. Here are the possible syntactic interpretations of such a phrase:

1: They [[[moved their office] from the town] to the capital].
2: They [[moved [their office from the town]] to the capital].
3: They [moved [[their office from the town] to the capital]].
4: They [[moved their office] from [the town to the capital]].
5: They [moved [their office from [the town to the capital]]].

Fig. 1 shows these five structures in a more demonstrative way using the Dependency Grammar notation which we consider more appropriate for our discussion. A native speaker would choose the structure 1 as the only possible interpretation by means of taking into account some additional information. As we will show, this information is of lexical and syntactic, but not semantic, nature, and a special dictionary is necessary for a parser to resolve this ambiguity.

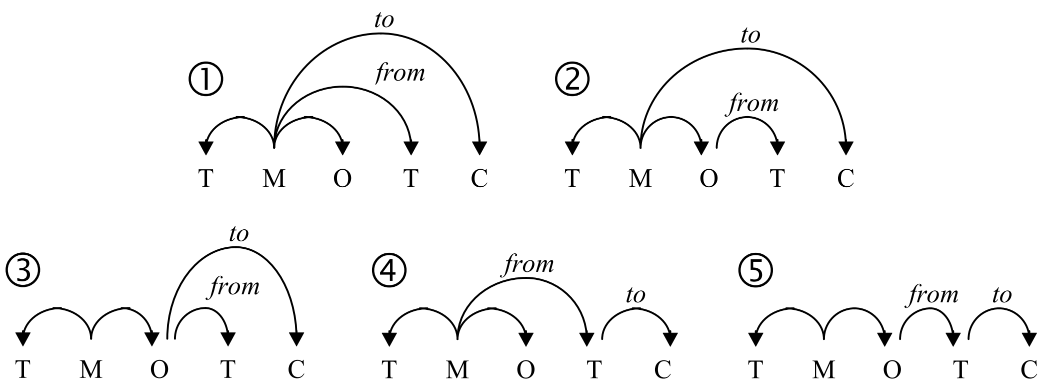

Fig. 1. Hypothetical syntactic structures for the sentence *They* (T) *moved* (M) *their office* (O) *from the town* (T) *to the capital* (C).

Such ambiguity is very common in syntax analysis. The number of variants grows exponentially with the number of prepositions [7], e.g., the phrase *They moved their office from the town in the North to the capital of the country* has 42 such variants of syntactic structure, etc.

This ambiguity cannot be resolved by general grammar rules related to the word order even taking into account specific prepositions. Indeed, all the five patterns shown on Fig. 1 are possible in English with the same prepositions; here are the corresponding examples:

1: They moved	their office	*from* the town	*to* the capital.
2: They told	the news	*from* the town	*to* the neighbor.
3: They prohibited	any movement	*from* left	*to* right.
4: They excluded	this word	*from* the preface	*to* the book.
5: They published	an excerpt	*from* the preface	*to* the book.

So the problem can only be solved by taking into account some *lexical* properties of the words. What is their nature? The example *They moved their office to the capital from the town* shows that these properties are not related with word order: the starting and ending points are still the same, though the word order is different. However these lexical properties are not of semantic nature. Indeed, if we know that the following phrases: **They moved their office from the dog to the idea* or **They sold a book to ten dollars for the customer* were written by a literate native speaker, we will have to admit that their syntactic interpretation is the pattern 1, and the semantic interpretation of the second phrase is that *ten dollars* is the buyer and *the customer* is the price, despite of absurdity of such a meaning[1]. On the other hand, with other prepositions and the same main words the phrase has another structure: *They moved the lawyer's office of the town near the capital* has the pattern 5, with the meaning of *move* 'to stir'.

A dictionary presenting the usage of specific prepositions with specific words is necessary to resolve the ambiguity of this kind. Since no semantic information has to be included in this dictionary, it is possible to learn this data from a large text corpus.

[1] In the meaning of Chomsky's *Colorless green ideas sleep furiously*.

3. Government patterns

3.1 Traditional government patterns dictionary

In [11, 14] a government pattern is defined as a table enumerating the syntactic valences, or actants, of the word, all the possible ways of their expression, and the limitation on their compatibility. Such dictionaries are intended in the first place for text generation and are also very useful for foreigners composing texts in the given language. For example, Fig. 2 shows a greatly simplified example of the entry for the word *move*$_1$ 'to change position', while other meanings are *move*$_2$ 'to excite', *move*$_3$ 'to request', etc. This example follows the pattern of the dictionary our group is developing for Spanish. In a Russian dictionary, instead of or together with the prepositions, the grammatical cases are indicated.

move$_1$	
Agent A transfers object O from starting point S to destination point D by the trajectory T.	
Way of expression	Example
A = 1: agent	
1.1. Noun (agent)	*a man / the Government* ~
O = 2: object	
2.1. Noun (object)	~ *their office*
S = 3: starting point	
3.1. *from* Noun (location) 3.2. *out of* Noun (location)	~ *from the town* ~ *out of the town*
D = 4: destination point	
4.1. *to* Noun (location) 4.2. *into* Noun (location) 4.3. *towards* Noun (location)	~ *to the capital* ~ *into a new apartment* ~ *towards the exit*
T = 5: trajectory	
5.1. *by* Noun (location) 5.2. *through* Noun (location)	~ *by the shore* ~ *through the forest*
Restrictions: none[2].	

Fig. 2. A government patterns dictionary entry.

The actants are supplied with the explanations of the semantic roles. The semantic marks like *location* or *agent* are intended to help disambiguation and also can be used in text generation if they are different within one actant. Such a dictionary enables a program that does not have any semantic information, to recognize the structure and the semantic roles in the

[2] Here we combine the transitive and intransitive meanings. For a transitive meaning only, the restrictions would mention that the second actant is obligatory.

phrases. It is also necessary for text generation or composition, by both a program or a person.

The knowledge on preposition usage, or, more generally, the ways of expression of valences, is language-dependent and because of this cannot be inferred by an algorithm. For example, for the second actant of the word ‘to marry’ English uses no preposition, Spanish *con* ‘with’, Russian *na* ‘on’. Thus, this information must be provided by a dictionary.

A traditional government patterns dictionary does not include the ways of expression of circumstances of the words, since this is not lexical knowledge, instead, these ways are fixed for the language in general. For example, in the phrase *They moved their office from the town to the capital at five o’clock on Monday for their convenience* the ways of expression of the circumstances do not depend on the main word *to move*.

In spite of great importance of such dictionaries, few attempts were made to compile one, or consistently provide the information on the ways of expression of valences in the common dictionaries. The dictionary [3] is the closest to this type for English. Manual compilation of such a dictionary is very labor-consuming. Thus, in our work on compilation of such Spanish dictionary we employ automatic learning the raw data from text corpora.

3.2 Government patterns for disambiguation

A simpler structure can be used for syntactic disambiguation and can be obtained automatically with the procedure described below. Fig. 3 shows an abridged example of such a structure[3]. The meaning of the first two columns will be discussed in the section 0. The symbol ‘∅’ denotes the empty preposition, i.e., the direct object, the symbol ‘—’ denotes the absence of any arguments. In this example we do not consider the first actant, the subject.

move			
p^+	p^-	Combination	Example
8892	3782	—	*Jill moved impatiently.*
3478	921	to	*John moved to the new apartment.*
372	123	∅ + from + to	*The firm moved its office from the town to the capital.*
135	342	∅ + out of	*She moved the table out of the room.*
83	58	∅ + into	*We moved the device into the house.*
76	782	to + for	*The family moved to the South for sake of the child.*
34	89	∅ + from + through	*He moved the table from the room through the door.*
30	219	to + at	*Jack moved to the new apartment at five o’clock.*
25	38	to + through	*She moved to the South through the forest.*
9	13	towards	*The group moved towards the mountain.*
1	463	of	*She moved the table of John’s friend.*

Fig. 3. An entry of the dictionary used for disambiguation[4].

[3] This is not the real output of the program. Since we worked with Spanish and Russian, we are using artificial English examples in this article. The numbers are chosen by hand an only as an illustration.

[4] The last line illustrates an error in the dictionary.

Fig. 3 shows the data that can be obtained automatically from the text with the procedure discussed in the section 0, along with the examples that also can be obtained automatically. There are significant differences with the traditional dictionary, Fig. 2:

- Only possible combinations found in the texts are shown in the table, the valences are not grouped together into actants.
- No information on the semantic roles is provided.
- On the Fig. 3, no semantic types of the words, like *agent* or *location*, are shown, however, if the information on the semantic types is available, it can be added to the table in the same way, e.g., "∅ (object) + out of (location)," though this would greatly enlarge the table.
- Some erroneous combinations may be present in the list, as it is shown at the last line of the table. However, the weights, see the section 3.3, of these combinations are usually very low, lower than the threshold of eliminating the combinations from the dictionary. Thus, they appear in the final list in very few cases.

On the other hand, the data set has additional information, and in the first place the statistical weights discussed in the section 3.3.

3.3 Statistical weights of patterns

With each combination, the number of its occurrences in the text corpus, denoted by p^+, is included. First, this number is shows the reliability of the information on this pattern. Second, in text generation or composition it allows to choose the most common way of expression of the actants. However, the main use of this number for disambiguation is discussed in the section 4. More precisely, this number is not exactly the number of occurrences, instead it is a weighted by the probability of each occurrence to be the true variant of the phrase structure, as it is discussed in the section 0. This is a technical trick, with the intended meaning of this figure being just the number of occurrences in the correct structures.

More interesting is the second field, p^-. This is the number of occurrences of the given combination in the *incorrect* variants of the phrase structure built by a specific parser. E.g., if the parser builds all the five possible variants for the phrase shown on the Fig. 1 and the corpus consists of only one phrase, then the pattern *town to* gets the values $p^+ = 0$ and $p^- = 2$, the latter from the variants 4 and 5.

This information is used for disambiguation: When a specific pattern is observed, is it more probably that this combination was found in a real structure or that it is built due to a mistake of the parser? Does the parser more frequently really detects this combination or mistakes something else for it? The disambiguation procedure is discussed in the section 4. The number p^- is also weighted by the probability of a specific variant to be false.

3.4 Semi-automatic compilation of the traditional dictionary

Though the data shown on Fig. 3 is intended primarily for automatic disambiguation, it is possible to use this raw data for semi-automatic compilation of a classic dictionary, such as

Fig. 2. We use a dialogue procedure. The algorithm of partitioning of the set of prepositions into actants for one entry works in the following steps:

1. The prepositions are grouped together so that no group contains two prepositions that belong to the same combination. These groups correspond to the hypothetical actants of the word.
2. Of all such possible partitions, the ones resulting in the minimal number of groups are chosen.
3. All the possible orders of the set of the groups are considered, with the restriction that the group containing the direct object must be the second actant. For each of such orders, a measure is calculated according to the word order in the combinations: The variants for which more combinations agree in the order with the ordering of the groups are scored better.
4. The ordered partitioning with the best score is presented to the user. The user can remove some of the prepositions as related to circumstances rather than to actants, or move a preposition to another group. After each user action, the calculations are repeated taking into account the restrictions introduced by the user, and an improved version of partitioning is presented to the user.

The process repeats until the user accepts one version or chooses to continue manually. At each stage, the user is presented with the examples, that are also included in the final dictionary.

After the actants have been determined, two kinds of hypotheses are presented to the user:

- The hypotheses on obligatory actants. If some actant is present in all the available examples, the program suggests to mark it as obligatory. The verbs with obligatory second actant are called transitive, such as *to give smth.* There are some verbs with two obligatory actants, such as *to tell smth. to smb.*: both phrases **He told this news*, **He told to Jack* are incomplete.
- The hypotheses on incompatibility of actants or individual variants of their expression, e.g., individual prepositions. Only pairs of prepositions are considered, and if two prepositions belonging to different actants are not found together in examples, the program suggests that they are incompatible. Since the number of such hypotheses is often very big, special heuristics are used to order them. However we have to admit that this feature was not very useful so far.

Though on the Fig. 3 only one example is shown, out program collects up to 10 examples for the same combination. They are chosen from the text corpus based on a compound criterion: (1) they cover the corpus approximately proportionally and (2) the examples are kept with the best scores assigned by the procedure described in the section 4. The examples are ordered by the latter scores and the best one is the first to be presented to the user, but the user can view all of them and choose the best one, remove some of them, search the corpus for other examples or enter a new one manually.

Of course, the semantic interpretation of the word and the semantic roles corresponding to the syntactic valences are added by the user manually.

4. Disambiguation with weighted government patterns

By disambiguation we mean assigning to the variants the weights according to the probability for the given variant to represent the correct structure of the phrase. This is better than just to choose one of the variants and reject the others. The weights can be used for ordering the variants. The variant with the greatest weight is considered first by the other modules of the system; if for some reason it cannot be accepted, the next variant is considered. Also the weights can be combined with other possible estimations of the correctness of the variants. Finally, these weights are used internally by our procedure as it is described in the section 0.

Let us suppose that a parser is available that builds for each phrase some variants of the syntactic structure. We suppose that the parser *always* builds the correct structure for a phrase and possible some additional incorrect variants. In this section we suppose that the government patterns data set, Fig. 2, is already given.

We consider the model of information transmission in the presence of the noise. The set of the variants generated for one phrase is considered a received package of signals. The signals are emitted by two different sources: the source S^+ produces only the correct variants, and the source S^- only incorrect ones. Each time we receive a package, exactly one signal in it was emitted by the source A and all the others, the noise, by the source B, see Fig. 4.

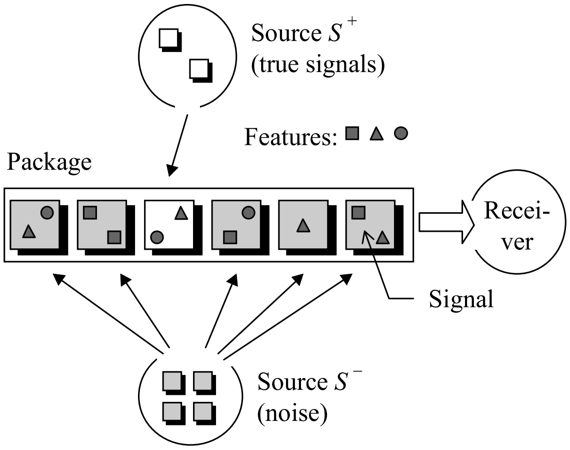

Fig. 4. Information package model.

There is a set of *features* of the signals; for each feature, a signal can have such a feature, or several occurrences of it, or none. In our case a feature is a specific combination of prepositions related with a specific word, the variant has this feature if in the given structure, this word is connected exactly with this set of prepositions.

The receiver can observe the features, and its task is to guess, which signal was issued by the "right" source S^+. The probabilities for each of the sources, S^+ and S^-, to assign a given feature to the signal that it issues are known, these are the values p^+ and p^-, correspondingly. We do not describe here the handling of morphological ambiguity that involves the frequencies of specific words and a little bit different handling of the probabilities p^+ and p^-. In fact the morphological ambiguity is very rare in the variants of parsing, since it is usually resolved by the syntactic grammar itself. Thus we suppose that all the variants include the same set of words and prepositions, but connected differently to each other, see Fig. 1.

Now it is easy to estimate the probability of the hypothesis H_i that the signal number i in the package is the true one, i.e., was issued by S^+, let's call this the weight w_i of the hypothesis H_i. We suppose that the features are numerous so that for each specific feature, the probability *not* to be included in a specific signal is near enough to 1. A simple reasoning based on the Bayes theorem proves that the weight w_i is a product

$$w_i = C \times p_i^{apr} \times \prod \frac{p_j^+}{p_j^-}$$

by all the features j found in the signal number i, where p_i^{apr} is the a priori probability of this hypothesis; C is the normalizing constant, since the total probability of the hypotheses H_k is to be equal to 1. The problem of division by zero never occurs due to the constant λ, see the section 5.

When using the data set extracted from a corpus to disambiguate other texts, some combinations are absent in the dictionary. In this case the corresponding factor should be set to a little value ε, since we believe that the probability for the parser to make an error of a new type is greater than the probability to find a new real combination. This value should be non-zero to allow the comparison by other factors. As it will be shown in the section 0, even while learning the probabilities from a corpus, the process speeds up considerably if the combinations with little quotient p^+ / p^- are eliminated from the dictionary. This policy agrees with the rule of using little values for not found combinations; the threshold for eliminating the combinations should be set approximately to ε.

Thus, the procedure for assigning the weights to the hypotheses of the syntactic structure of one phrase is as follows:

1. All the variants permitted by the grammar used by the parser are built, these are the hypotheses H_i.
2. Any available knowledge and procedures are applied to estimate a priori the "quality" p_i^{apr} of each hypothesis. These procedures can take into account, let us say, the length of the links (the shorter links are generally scored better), semantic coherence of the structure [9], weights of the grammar rules used in it [1, section 7.6], etc. If no information of such kind is available, equal weights are assigned to all the hypotheses.
3. For each variant of the syntactic tree, the features of this variant are looked up in the dictionary. In our case, for each word, the combination of prepositions attached to this word in the current syntactic tree is retrieved from the list. If the combination is found, the weight w_i of the variant is multiplied by its p^+ / p^- , otherwise it is multiplied by ε.
4. The weights w_i are normalized so that $\Sigma\, w_i = 1$ for the variants of the structure of the same phrase.
5. The variants are ordered by the weights w_i, and the variant with the greatest weight is considered the result of the analysis.

Some generalizations of the method will be discussed in the section 7, but the changes only concern the nature of the features operated upon and do not concern the procedure itself.

5. Learning the weighted government patterns from a corpus

Now let's consider the opposite situation: There is a parser and a disambiguation procedure that assigns the weights to the hypotheses; the frequencies of occurrence of each feature in the correct and wrong variants are to be found. In our case a feature is a words along with the set of arrows leading *from* this word, taking into account their types, see Fig. 5.

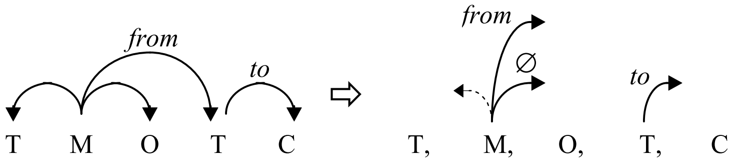

Fig. 5. Features of the syntactic tree.

If the procedure directly chose the correct variant, the only thing to do would be to increment the counter p^+ for all the combinations found in the correct variant, and p^- for the combinations found in the incorrect one. Since the disambiguation procedure only determines the weights of the variants, we can consider the same model, Fig. 4, and again apply the Bayes theorem. For each variant, the probability that it was issued by the "right" source S^+ is w_i and the probability that it was issued by the source of noise S^- is $1 - w_i$; this values are accumulated. To calculate the mean values, the total should be divided by the number of signals generated by the sources S^+ and S^-. Let V be the number of variants generated by the parser and S the number of sentences in the corpus, then the total number of the correct variants is S and of incorrect is $V - S$. Thus the formulae are the following:

$$p_j^+ = \frac{\sum w_k}{S},$$

$$p_j^- = \frac{\sum (1 - w_k) + \lambda}{V - S},$$

where the summation is done by all the occurrences of the feature j in the variants w_k, and the meaning of λ is described below.

The formulae work for the ideal case, when the corpus is so big that any possible type of combination or error occurs many times during its analysis. In reality, due to infinite variety of the constructions in open texts, all the possible words and combinations cannot occur in any corpus, even large enough, and what is more, vary numerous are the cases that occurred in the corpus very few times, or even one time. Such cases introduce great instability in the model since the quotient p^+/p^- for them is either very big or very small, and this value is almost random, since each additional occurrence would greatly affect it[5]. There are different

[5] Though in [8] the significance of rare cases is especially emphasized, we did not observe such an effect, so that smoothing of the rare cases gave a much better results.

methods to suppress such rare cases in the statistics. We have chosen to artificially add some number λ of occurrences of each combination in the *false* variants; the experiments have shown that this method works best. The value of λ was also chosen experimentally, it turned out that the best results are achieved with $\lambda = S$.

Thus, the procedure for accumulating the statistical weights for the combinations is as follows:

1. All the variants of the structure are built for each phrase of the corpus.
2. The variants for each phrase are estimated, i.e., are given the probability weights w_i such that $\Sigma\ w_i = 1$ for each phrase.
3. For each combination found in each variant, the counters p^+ and p^- are incremented by the values w_i and $1 - w_i$, respectively. The initial values are zeroes.
4. At the end, the value of $\lambda = S$ is added to each p^- and the values are divided by S and $V - S$, respectively.

After the values are determined, the combinations with the quotient p^+ / p^- less than some threshold value ε can be eliminated from the dictionary, as it will be described in the section 6. To speed up the procedure, the after the step 2 the variants with the probability w_i less than some threshold can be ignored.

6. Iterative disambiguation and learning

In the sections 4 and 5, the procedures are described working in the mutually opposite directions. For disambiguation of a large enough corpus, as well as for initial training of the model, these two procedures can be used iteratively. The work starts from, let us say, an empty dictionary. The disambiguation procedure then assigns equal weights to all the variants, or keeps their a priori weights. At the next step, these weights are used to train the model, i.e., to determine the frequencies p^+ and p^-. Then the process is repeated. As experiments show, the iterative process converge very quickly.

Of course, the information does not appear from nothing. Throughout the corpus, the variants having the same combinations are “connected” to each other in the model, they either “help” to each other to win the competition within their sets of variants for one phrase, or suppress each other when they loose this competition. Thus the model optimizes itself to the state when the winners in each set have as much as possible in common. Since the sentences are different, the errors are random, and at the same time the grammatical sentences have some combinations in common, thus, these sentences, all together in the corpus, tend to win the competition.

Once the set of the variants and the set of combinations found in these variants have been built, the data structures used in the iterative procedure can be fixed in the computer memory, because all the operations in the procedures are arithmetical and do not produce any new objects. However, depending on the implementation, the time of access to the dictionary can be significantly reduced at the later iterations by eliminating the combinations with the value of p^+ / p^- less than a threshold or by ignoring the variants with the weight less than some another threshold. After the first iteration, the dictionary is usually very big, but after two or three iterations, nearly only the correct combinations are left in the dictionary, that greatly reduces its size.

7. Experimental results

In our experiments, two values were measured: the similarity between the dictionary built by the program and the "true" one, and the percentage of the correctly parsed sentences. By correctly parsed sentences we mean the ones for which the variant with the highest weight was the true one. The techniques of experiments to measure these two values were different.

To measure the similarity between the dictionaries, a real text corpus could not be used because the "true" dictionary that the persons that wrote the texts had in mind is unknown. Thus, to check the methods, we modeled the process of text generation to obtain a quasi-text corpus built with a known dictionary [4]. Only the statistical characteristics of the text were modeled, such as the length of the phrase and, of course, the preposition usage; we paid the main attention to the constructions common for Spanish and Russian. With this method we could measure the percentage of the correctly parsed phrases as well.

In various experiments we observed all the three patterns of convergence mentioned in the similar context in [10], depending on the formulae and parameters we used, as well as on the size of the corpus. In the *initial maximum* pattern, the dictionary obtained after the first iteration, i.e., with equal weights of the variants, was the best, along with the percentage of the variants guessed correctly with this dictionary; at the subsequent iterations, both estimations were getting worse. In the *early maximum* pattern, the best values were achieved after several iterations, and then they slightly degraded. Finally, with the formulae described here and the parameter λ of the order of S, as described in the section 5, the *classical* pattern was achieved: the values tended to grow and quickly stabilize at the relatively high level. However, even in this case we observed slight elements of the *early maximum* pattern: after reaching the maximum, the percentage of correctly guessed variants fell insignificantly, usually within 1%, and stabilized at that value.

To as a measure of nearness between the two dictionaries, the built one and the "true" one, we used several measures: the percentage of incorrect combinations, the coverage, and the difference of the probabilities of usage p^+ for the correct combinations. After few iterations these values stabilized at the level around 5% of incorrect combinations and 80% of similarity of probabilities. The coverage was not good enough in our experiments (about 30%) since due to the technical limitations of our program so far we used relatively small corpus.

A typical sequence of the percentages of the correctly guessed variants was 37%, 85%, 89%, 90%, 90%, etc., or, taking into account only the phrases for which the parser generated more than one variant, 16%, 80%, 86%, 87%, 87%, etc. Then the results stabilized. The first figures in both sequences were obtained with the equal weights, by picking just an arbitrary variant for each phrase. The last figures of the sentences show the accuracy reached with the method being discussed.

The second set of experiments was carried with real Spanish and Russian text corpora. As a Spanish corpus we used mainly the texts kindly provided to us by the publisher of Gazeta UNAM, the newspaper of UNAM University, Mexico City; the corpus contained approximately 8 millions words. We used a very simple context-free parser to build the initial set of the variants; the grammar contained 41 rule in a special language. A spot check of the results showed good convergence of the method with the best value reached so far being 78% of correctly parsed phrases; on unseen data analyzed with the dictionary built at the training stage, this figure so far was 69%. Note that all the figures reflect not the number of correctly attached prepositions, but instead the number of correctly parsed phrases, so that if any part of the phrase was not parsed correctly, the whole phrase was considered parsed wrong. With a more elaborated grammar, we expect to reach better results.

Surprisingly, in the experiments we did not observe any advantage of using some nontrivial initial values for the weights of the hypotheses or for the dictionary. The best results were obtained with equal weights, i.e., with initially empty dictionary. However, the a priori information can be used at each iteration, as it was described in the section 4.

8. Generalizations and future work

The method has many possible variants. For example, different kinds of information can be taken into account in the list of combinations. If there is any lexical information available from the parser, such as part of speech (in Spanish prepositions can introduce verbs), animateness (in Russian this is a morphological property), semantic class (such as *person*, *agent*, *living being*, *organization*, *object*, *action*), etc., they can be added to the government patterns, provided that the corpus is large enough, due to the sparse data problem. In this case some method of merging the patterns of similar structure, but with different characteristics of the governed word should be applied. For example, if two patterns have comparable weights and differ only in animateness of one of the valences, they should be merged in a common entry without the animateness mark.

On the other hand, counting each preposition separately will very significantly reduce the sparse data problem. For example, instead of one pattern *move* + *from* + *to* + *through* three independent patterns can be considered: *move* + *from*, *move* + *to*, *move* + *through*, though this may reduce the accuracy when the model is trained on a large enough corpus.

Other generalizations concern the very nature of the objects for which the statistics is gathered, this is why throughout the paper we preferred to refer to them as to abstract *features*. First, by features the grammar rules used in the parsing process can be considered. This will turn the grammar used by the parser into a stochastic grammar [1]. Second, we expect that with a vary large corpus, a similar approach can be applied to word combinations, for both syntactic disambiguation and composition of the dictionary useful for human readers [5]. All the three methods, namely, based on the weights of the government patterns, grammar rules, and word combinations can be used simultaneously as described in the section 4.

Finally, the method can be translated into the language of neural networks. Indeed, the variants of the parsing can be viewed as neurons, a features common to two variants can be viewed as a mutually exciting link, while any two variants belonging to the same phrase can be viewed as mutually inhibiting; disambiguation is viewed as excitement of exactly one neuron in each set of the mutually inhibiting ones. We plan to investigate whether the neural network techniques can increase the performance of the method.

9. Conclusions

A procedure of syntactic disambiguation based on the use of syntactic government patterns the statistical weights is proposed. The government patterns and their weights can be automatically learned from a large text corpus and can be used for disambiguation of other texts. The method has the following advantages:

- No hand preparation is required to train the model. However, a morphological and syntactical analyzers are required since the method is devoted to disambiguate the results of such parsing.
- The method is compatible with other methods of disambiguation, especially with methods that produce an estimation of probability for each variant.
- The method is tuned to a specific parser, taking into account the balance between the correct and wrong assignments of prepositions to words.
- The data built by the algorithm is lexical, so that the amount of processed data does not increase with the growth of the grammar.
- The data set learned from the corpus is useful for semi-automatic compilation of a traditional government patterns dictionary, that is used both for semantic analysis and by the foreigners composing in the given language.
- The government patterns used by the method correspond to some linguistic reality unlike, let us say, the probabilities used by the Hidden Markov Model.

We believe that the latter means that the native speakers are aware of such a reality and in text composition intentionally try to avoid constructions that would be misleading with respect to government patterns of the words, cf.: $^?$*They laughed at this place* vs. *They laughed here*, $^?$*He spoke with the director of the new plan* vs. *He spoke of the new plan with the director*.

References

1. Allen, James. *Natural Language Understanding*. The Benjamin/Cummings Publishing Company, Inc., 1995.
2. Baum, L.E. *An inequality and associated maximization technique in statistical estimation for probabilistic functions of a Markov process*. Inequalities, 3:1–8, 1972.
3. Benson, M., E. Benson, and R. Ilson. *The BBI Combinatory dictionary of English*. John Benjamins Publishing Co., 1986.
4. Bolshakov I.A., A.F. Gelbukh, and S. Galicia-Haro. *Simulation in linguistics: assessing and tuning text analysis methods with quasi-text generators*. Proc. of International workshop on computational linguistics and its applications, Dialogue-98, Khazan, 1998, Russia.
5. Bolshakov I.A., P.J. Cassidy, and A.F. Gelbukh. *CrossLexica: a dictionary of word combinations and a thesaurus of Russian* (in Russian). Proc. of International workshop on computational linguistics and its applications, Dialogue-95, Khazan, 1995, Russia.

6. Brill, E., and P. Resnik. *A rule-based approach to prepositional phrase attachment disambiguation*. Proc. of ACL, 1994, Kyoto, Japan.

7. Church, K., and R. Patil. *Coping with syntactic ambiguity, or how to put the block in the box on the table*. American Journal of Computational Linguistics, 8 (3 – 4):139 – 149.

8. Collins, M., and J. Brooks. *Prepositional phrase attachment through a backed-off model*. Proc. of the Third workshop on very large corpora, 30 June 1995, MIT, Cambridge, Massachusetts, USA.

9. Gelbukh, A.F. *Using a semantic network for lexical and syntactical disambiguation*. Proc. of CIC'97, Nuevas Aplicaciones e Innovaciones Tecnológicas en Computación, Simpósium internacional de computación, CIC, IPN, Mexico D.F., 1997

10. Elworthy, D. *Does Baum-Welsh re-estimation help taggers?* Proc. of Fourth Conference on Applied Natural Language Processing, Stuttgart, Germany, 1994.

11. Mel'čuk I. A. *An experience of the theory of Meaning ⇔ Text models* (in Russian). Nauka, Moscow, 1974.

12. Merlo, P., M. Crocker, and C. Berthouzoz. *Attaching multiple prepositional phrases: generalized backed-off estimation*. Proc. of Second conference on empirical methods in natural language processing (EMNLP-2) August 1 – 2, 1997, Brown University Providence, Rhode Island, USA.

13. Pereira, F., and Y. Schabes. *Inside-outside reestimation from partially bracketed corpora*. Proc. of ACL, 28 June – 2 July, 1992, University of Delaware, Newark, Delaware, USA.

14. Steel, James, ed. *Meaning – Text Theory. Linguistics, lexicography, and implications.* University of Ottawa press, 1990.