KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 50–57

SYNTAGMA PROCESSING FOR INCOMPLETE ANSWERS

ADRIAN ONET[(1)]

Abstract. By trying to find a solution to incomplete answer processing, answers that are very frequent in a usual communication scenario based upon question-answer pattern, we developed an algorithm able to reconstruct the incomplete answer by using the question syntactical environment. Thus, one of the problem related to natural answers are the syntagmas. We call syntagma an incomplete answer that resumes to a phrase not to a grammatically correct sentence based on a subject and a verb. For example, if we consider the question "*What is your favorite color?*", most of the answers will be of the following form "green". Unfortunately, such an answer can't usually be processed by using an English grammar. In our SPEL (Syntactic Parser for English Language) system, we have introduced an algorithm that is able to reconstruct the answers from the given syntagma and the initial question, without affecting the semantic information given by the answer.

1. Introduction

In a usual communication scenario that necessarily involves a question-answer pattern the most common situation that we have to resolve is the syntagma answers. This means that all the incomplete answers that are received to a given number of questions must be reconstructed by using the specific syntactical structure of the question.

In order to eliminate the irrecognizability of this kind of incomplete sentences, we will present in this paper an algorithm for syntagma reconstruction by using the user's incomplete answer to a question and the respective question. The algorithm is capable to create an answer that is syntactically correct. It will consequently have a subject and a verb that respect the basic syntactical pattern. The presented algorithm is used as part of the SPEL system and it was tested on more than 40000 answers with promising result. However, the algorithm is not fully proved but it has a good rating of reconstructing the correct answer. Another benefit of this algorithm is the fast processing: it uses only some of the semantic and syntactic

2000 *Mathematics Subject Classification.* 68T35, 68T50, 91F20.

Key words and phrases. Syntagma Processing, Natural Language Processing, Incomplete Sentence Reconstruction.

information from the question and from the answer. First we will do a short introduction of the SPEL system that incorporated this algorithm, afterwards, we will present the necessary steps to implement the algorithm itself and its further improvements.

2. SPEL Description

The SPEL is designed to be able to syntactically parse English complete and incomplete sentences. The system is based on a DCG grammar and an extension of the Wordnet [1] dictionary. Before we go any further into the description of our system, we have to enhance the fact that there are many syntactic parsers in the literature, from these we can mention:

(1) AGFL [2] that is based on a two layer gramar;
(2) *Link Grammar Parser*, based on linked grammars [3];
(3) RASP (*Robust Accurate Statistical Parsing*) system, based on a statistical analysis of the lexical information [4];
(4) *Connexor* [5] which is also based on a statistical parsing model.

As opposed to these systems, SPEL is based on a strict grammar that is unable to recognize syntactically incorrect sentences or correct sentences that do not have their corresponding rules in its grammar. Its efficiency strictly depends on the grammar that we have built which allows us to detect the syntactical correctness of the processed sentences that are processed. However, despite these, our system presents the following advantages, by providing:

- the deep analysis of the syntactically correct sentences;
- the extensibility and the modularity of the grammar;
- the possibility of inserting semantic rules over the existing rules for a semantic parsing;
- a flexible adaptability of the grammar and the possibility of constructing a new system for automatic sentence translation.

The disadvantages of SPEL over the existing systems concern mostly, on one hand, the processing time and, on the other hand, the morphological, syntactical correctness of the words in the sentence. But some of these disadvantages were already considered to be incorporated and will be eliminated in future releases. As we already stipulated, one of the problem that SPEL may encounter is that the processing time for some complex sentences can be very long. This usually happens for sentences that contain polymorphic words, such as a verb which has the same form as the corresponding noun (*to work / work*) or an adjective and an -ing verb form (such as *interesting*). This problem occurs mainly because of the size of the dictionary. On the other hand, SPEL will not recognize syntactically incorrect sentences. That is sentences that do not respect the static rules from its DCG Grammar. This, in fact, ensures the fact that the system depends strongly on the syntactical correctness of the sentence. Also, the system doesn't accept

elliptic words or ortographically incorrect words. As the grammar is implemented in Prolog, the order of the rules is the order the sentence will be matched. Thus, the first match will be considered the desiderate parse. But this is not always the case if we consider the polymorphic words, where we can not decide, even by statistical choice, which morphological value of a word is to be considered first.

As an improvement for SPEL, we want to combine the existing grammar with the linked grammars, in such a way that it will also introduce semantic elements in the syntactic parsing. This will improve the system by choosing the morphologic occurrence of a polymorphic word (for example, the word "living" can be adjective, noun and -ing verb) that is most semantically appropriate with the sentence context. Another benefit of such an approach is that the parsing time can be considerably reduced as the grammar will use only one morphological value of a word, avoiding checking irrelevant paths.

Also, The SPEL architecture is based on English grammar written in a DCG form, the grammar is interpreted by a Prolog engine. The dictionary is an extension of the Wordnet dictionary and is stored in a relational form. The extension from the Wordnet is the adding of more syntactic information for the words contained in Wordnet. To analyze a certain sentence, SPEL first selects from the dictionary the words that may contribute to the sentence. The next step is to call the Prolog engine with the given words from the dictionary and the sentence to be evaluated to try to do the matching. If the match is successful, SPEL is able to draw the resulted syntactic parse. As regarding the incorrect sentences, the system is capable to recognize the incorrect words.

One of the system usages is to parse users' answers to psychological tests and return statistics of the morphological parts discovered in the users' answers. One of the issues with these answers is that the users tend not to answer in a sentence to the given question (for example, *What is the emotion that you feel when looking at the inkblot?*) or question-task (of the form *Describe what activity could be taken place in this sequence*), but rather to give only a syntagma. For example, for the following question "*What did you have for lunch?*" the user will answer in a syntagma like "*a donut*". Such a syntagma will not be able to be parsed by the system. In the following, we will give an algorithm able to transform these syntagmas in correct sentences using semantic and syntactic information from the question and syntactic information from the answer.

3. Syntagma processing

3.1. **Syntagma problem.** As we mentioned in the previous paragraph, the users tend to answer to a question in sentences that are not syntactically complete, most of the time the user answer is a very short syntagma that answers to the question. The goal for the SPEL system is to provide a system which could be able to recognize the syntagmas and to be able to reconstruct a syntactically correct

sentence from them without affecting the semantic information which remains intact.

3.2. **Identifying syntagmas.** One of the challenging problems regarding syntagma processing is, in fact, the syntagma identification. As for now the SPEL system considers syntagmas all the sentences that are not correctly identified by the grammar. The problem with this approach is that the system will consider as syntagma even the answers that are not syntactically correct (according to the given grammar). To avoid considering all the failed sentences as syntagma the system eliminates from syntagma the answers that could not be categorized in the Syntagma Categorization step (see 3.4). In the current state, the SPEL system first applies the answers against the grammar. In the case that the sentence is not recognized as being correct, the system will try to categorize the answer as a syntagma category. If the syntagma can be categorized than the sentence is considered as syntagma and the syntagma resolving step will occur that will reconstruct the sentence from the syntagma and the information from the question deconstruction step, finally the new reconstructed is applied again against the grammar. If the new reconstructed answer is recognized by the grammar, then the sentence is considered correctly reconstructed, otherwise the sentence is not recognized as a syntagma. The downside of this approach is that an answer has to be processed twice against the grammar doubling the processing time.

3.3. **Question Deconstruction.** In order to process the recognized syntagmas from the previous step, we need to determine some syntactic and semantic information for each question, information that was involved in the syntagma answer. To do this we will construct an array of pairs of properties of the form attribute:value. Depending on the scope of the questions, there are properties that need to be included. In our case, we considered the following attributes for each question:

- **Syntactical Subject**: representing the subject to which the question refers;
- **Verb involved**: representing the verb that contributes to the answer construction (most of the time it is part of the question);
- **Verb preposition**: sometimes the verb that contributes to the answer needs a preposition, as for example "of". This preposition will be given by the value of this attribute;
- **Logical Subject**: this attribute represents the object in the question. The object in the question becomes the action agent in the answer. The answer may have a syntactical subject but the real agent will be the value given by this parameter;
- **Original syntactical subject**: this is usually the second subject from the question;
- **Question verb**: the exact form of the verb that is also part of the answer;

Depending on the questions domain (for example, psychological test related question), there can be added other specific properties.

Let us consider the following question-task: *Describe what activity could be taken place in this sequence.* In this case, the syntactical subject is "*it*" resuming "*the activity*" expressed in the question, as the subject to which the question refers; the verb involved in this case is "*could be*"; the logical subject will be "*someone*" ("*someone*" is the agent of the action involved in the question); the original syntactical subject is "*I*"; the question verb in our question is "*could*"; the verb preposition is missing in this question. Finally, the properties array for the question will be represented as follows:

$$[syntactical\_subject(it), verb\_involved(couldbe), logical\_subject(someone), \\ original_s yntactical\_subject(I), question\_verb(could), verb\_preposition()]$$

In the following part of the article we will show how this information can be used in the sentence reconstruction from the syntagma that answers to the question.

3.4. **Syntagma categorization.** In order to be able to reconstruct the sentence using the syntagma, we will need, beside the question deconstruction, to categorize the syntagmas. This step is necessary in order to be able to apply specific rules for each kind of syntagma. We will present here only a partial question qualification that applies to psychological test answers. Thus, depending on the domain of the question, the syntagma classification involves several categories. Here are the main categories used by the SPELL system:

- **Participial syntagmas** - these are the syntagmas composed by a present participle (the forms in -ing). The syntagmas are considered part of this category if they start with a present participle verb. For example *having fun*, *looking at the sky* and *climbing a mountin*;
- **Subject elliptical verb syntagmas** - are the syntagmas constructed around a regular verb. To recognize these syntagmas, these are the ones that start directly with a verb, for example *work hard to get where I want*;
- **Noun phrase syntagmas** - are the syntagmas that represent a noun phrase. To identify these syntagmas, we have to pay attention to the structures that, if alone, are recognized as a noun phrase, for example: *a yellow building*;
- **Auxiliary verb elliptical syntagmas** - are the syntagmas that contain a present participle verb or a past participle verb and, also, where the previous word is not an auxiliary verb. There is a problem in order to do the classification of these syntagmas, because the verb form in the past participle is the same with the preterit form of the verb, so the confusion may occur between a subject elliptical verb or participial and auxiliary verb elliptical syntagmas;

All the answers that could not be classified here are not considered as syntagmas, but rather as syntactically incorrect sentences. This method is not an exhaustive method, but it can give very good results for domain specified questions (where most of the syntagmas tend to respect the existing rules). Another problem that arises here is the time needed to determine the syntagma category. In most of the cases, except for the noun phrase syntagmas, there is only a word lookup in the dictionary. And even more, if we consider that the sentences were previously checked against the grammar and the word retrieved from the dictionary, we already have the words morphological value, so the dictionary access is not needed in order to make the categorization. The same applies for the noun phrase syntagmas, as we can use the previous parsing phase to determine if the answer is actually a noun phrase. By using both this classification and the question deconstruction, we are now able to rebuild the sentences resumed by the syntagmas.

3.5. **Resolving syntagmas.** In order to reconstruct the sentences by using the question deconstruction and the syntagma characterization, we will create rules that apply for each syntagma category. As in the previous cases, these rules depend on the question domain and can't be used as a general rule for a particular syntagma category. Also, because of this, our solution doesn't provide a precise sentence reconstruction. Still from our practical result this algorithm gives a good ratio of well constructed sentences. Another problem represents the fact that the reconstructed sentence needs to be again applied against the grammar to check if it is a syntactically correct sentence. However, by building a sentence using the elliptical structures that we call here "syntagmas", we have the possibility to include in our parsing structures that usually are considered to be syntactically incorrect because elliptical. In the following we will present the rules used by the SPEL system in order to reconstruct the sentence from the syntagmas.

a) In the case of **participial syntagmas**, the sentence will be reconstructed using the following formula:

$sentence = syntactical\_subject + verb + logical\_verb + verb\_preposition + syntagma$

To demonstrate this, we consider the question-task from the section 3.3: *Describe what activity could be taken place in this sequence?* The answer that was given to this kind of question: *Having on a costume going to a Halloween party.* By using the given rule and the question deconstruction, the new recognized sentence will be: *It could be someone having on a costume going to a Halloween party.*

b) In the case of **subject elliptical verb syntagmas**, we use the following formula:

$$sentence = syntactical\_subject + syntagma$$

In order to exemplify this situation, we could have as an answer a subject elliptical verb syntagma, a construction of the type *looks like someone is crying.* According to our formula, our syntactical_subjet is it, so the rebuilt sentence will be: *It looks like someone is crying.*

c) For **participial syntagmas**, the formula to be applied will have the following form:

$$sentence = syntactical\_subject + verb + logical\_verb + syntagma$$

As an example of such syntagma, let us assume that for our question-task *Describe what activity could be taken place in this sequence?* the answer is *cared away by his emotions*. In this case, the reconstructed sentence will be: *It could be someone cared away by his emotions*.

d) To pursue our syntagma reconstruction examples, in the situation where the answer qualifies as a noun phrase syntagmas, the formula to be applied wil be:

$$sentence = syntactical\_subject + verb + logical\_verb + syntagma$$

Thus, as an answer to the same question-task as previously, the answer could be of the form: *a war scene with guns*. In order to reconstruct a syntactically correct sentence, the noun phrase syntagma will be consequently transformed as: *It could be a war scene with guns*.

e) For the situation where the answer is an auxiliary verb elliptic syntagmas, the formula will be:

$$sentence = syntagma\_subject + verb + logical\_verb + remaining\_syntagma$$

If we consider the response: *someone having a crisis*, the syntagma subject is *someone* and the remaining_syntagma is *having a crisis*. Here, the noun phrase preceding the participle becomes the actual subject of the reconstructed sentence, the verb involved in the question (in our case could be) becomes the main verb. Since the verb to be is an auxiliary verb, the participle in our syntagma is going to complete the verb, thus the solution: *Someone could be having a crisis*.

4. Conclusion

As we mentioned in this article, the given solution is not a precise solution, but it gives good results for domain specific sentences. As an improvement we can use multiple rule assignations for each syntagma category, each rule with a probability value assigned to it. In this case the algorithm will be changed in the sense that, instead of trying only one rule for the sentence reconstruction, it will try all the rules and select the one assigned with the highest probability. Another aspect that was not discussed in detail is the cost of this algorithm. The cost is a significant point as this kind of algorithms are mostly used in fields where there are a few questions answered by thousands of students. As it can be noticed, the SPEL steps do not involve a high cost, compared to the grammar application against the sentence. Still, a big cost comes from a second checking of the reconstructed sentence against the grammar to be sure that the reconstructed sentence respects the grammar. We are developing at the present time another version of the algorithm that involves all the processes to be fulfilled in the first grammar checking phase, by adding extra information to each participating word.

But in this case we have to consider not increasing too much the cost for the syntactically correct sentences. As mentioned before, this solution is used in the current SPEL implementation with very promising results.

References

[1] Ch. Fellbaum, Wordnet. An electronic lexical database, The MIT Press, Massachusetts, 1999
[2] C. H. A. Koster, E. Verbruggen, "The AGFL Grammar Work Lab", in Proceedings FREENIX/Usenix, 2002, pp 13-18
[3] D. Bechet, "K-valued link grammars are learnable from strings", in Proccedings of the 8th conference on Formal Grammar (FGVienna), Viena, 2003
[4] E. Briscoe, J. Carroll, "Robust Accurate Statistical Annotation of General Text", in Proceedings of the Third International Conference on Language Resources and Evaluation, Las Palmas, 2002, pp. 1499-1504
[5] The connexor parser wbe page: http://www.connexor.com/demo/syntax/
[6] P. Blackburn, J. Bos, Representation and Inference for Natural Language. A first Course in Compuational Semantics. Volume I Working with first order logic. Computerlinguistik, Universitt des Saarlandes, 1999
[7] A. Onet, D. Tatar, "The semantic representation of Natural Language sentences. A theoretical and practical approach", in PC 132 God, nr.1797, 2001, Budapesta pp.195-204

(1) Babes-Bolyai University Cluj-Napoca, Faculty of Mathematics and Computer Science
E-mail address: adrian@cs.ubbcluj.ro