Semantic Relations for an Oral and Interactive Question-Answering System

X, Y, Z

[1] name of an institute
[2] name of the next institute

1 Introduction

Semantic level processing is currently considered a promising way to significantly improve the performance of Question-Answering (QA) systems.

In this paper, we present the first steps of a work which aims to automatically detect semantic relations in user queries within the framework of XX, an oral and interactive QA system.

After a short description of XX and its current utterance analysis (section 2), we describe our objectives and the methodology we have chosen to represent and detect semantic relations in the specific context of that QA system (section 3). Then we describe what is currently implemented of this detection and give some prospects for our future works (section 4).

2 The XX system utterance analysis

The XX project aims at integrating a spoken language dialogue system and an open-domain information retrieval system in order to enable a human to ask a general question and to refine his research interactively.

The dialogue and QA modules of that system are in part based on *non-contextual analysis* the aim of which is to extract, from both user utterances and documents, what is considered to be *pertinent information*. The output of the analysis is twofold: a chunking which groups together series of words with coherent meanings and the typing of these chunks. The types can belong to several categories: named entities (person, location, time, organisation...), linguistic entities (e.g. verbs, prepositions), or specific entities (e.g. scores, colors...). There currently are 266 categories. The analysis is robust to spoken language, including automatic speech recognition output, and written language. The mean time per utterance or sentence is roughly 4ms. Figure 1 gives a classification of the used tags.

Figure 2 shows an example of such an analysis.

named entities	<org> *NIST* </> <eve> *festival Cannes 2007* </> <cit> *veni vidi vici* </>
indistinct entities	<Eve> *Cannes festival* </> *the* <Pers> *president* </>
extended entities multi-levels	functions, titles (bishop, president, professor, ...) colors, animals...
hierarchical super-classes	bishop → religious function → hierarchical function
thematic markers	<literature> *novels* </> <sport> *tennis* </>
inquiring markers	<Qqui> *who* </> *has ...* <Qmesure> *How many* </> *days*
interaction markers	<DA_close> *goodbye* </> <DA_yes> *yes please* </>
compounds	<NN> *data base* </>
verbal chunks	*they* <action> *take part* </> *to...*
linguistic entities	<stat_objet_plus> *the biggest* </> *exporter* *it* <adv> *often* </> *occurs*

Fig. 1. Entity types

<_Qneg_dial> **je ne veux pas d' informations** </> <_neg_info> <_prep> **sur** </>
<_pers> **Benedetti** </> </>
<_Qdial> **je voudrais** <1> **une** </> **information sur** </>
<_det> **le** </> <_range_objet> **dernier** </>
<_prix> <_Prix> **prix Nobel** </> <_type_prix> **de la paix** </> </>

Fig. 2. Annotation of a user utterance: *je ne veux pas d' informations sur Benedetti je voudrais une information sur le dernier prix Nobel de la paix* (*I am not looking for information about Benedetti, I want information on the last recipient of the Nobel Peace Prize*)

3 Objectives and methodology

The XX QA system is entirely based on the presence in the documents of typed chunks common with the query. Candidate answer scoring is computed from the proximity with these chunks. Our hypothesis (shared with other authors [CUI,BUC]) is that adding relations between chunks will give us a better score quality than simple proximity. Two main classes of relations exist: syntactic and semantic. We chose the latter. Ultimately, the scoring should be improved by unification between the relations found in the query and the documents.

3.1 Related work

In most works related to the search of semantic relations in QA systems, the search is based on the syntactic analysis of whole sentences. The detection of the semantic roles [NAR], [PRA] is an example of such an approach: a semantic role is a classification of a nominal phrase according to its relation (subject, direct object...) with the verb. Such relations are in practice more syntactic than semantic. Moreover, detecting them can be hard for text documents [GRI] and tends to break down on spoken queries.

Other works aim at detecting true semantic relations between chunks of the sentences. Usually, these relations are between nominal phrases (hyponymy, hyperonymy, etc.) [HEA] [GIR]. While very interesting, they are mainly useful in extracting semantic lexical knowledge and are currently too specific for our purposes.

Therefore, we have decided to try new approaches, specific to our problems.

3.2 Representation and detection of the semantic relations

We have chosen to represent semantic relations as logical formula over the chunks on the annotated utterance.

This formula is built from predicates with arguments linked with logical connectives.The name of the predicate gives the nature of the relation and the arguments refer to a list (which can potentially be empty) of chunks of the utterance.

To be useful, the predicates should represent a reasonably simple and generic semantic relation. This will help unifying queries and documents which a larger number of more specific predicates would impede.

Currently, the detection of the relations is split into three steps. All of these are heavily dependent on the previously detected entities.

1. **Grouping of some of the chunks into syntactic groups**, in particular nominal and verbal phrases. Reducing the number of chunks that must be dealt with makes the following steps easier; moreover, linking grammatical words to the content word to which they are linked is a way towards disambiguation. Examples of syntactic groups are given in Figure 3 (prepositional nominal phrase, <PNG_on>) and in Figure 5 (verbal phrase, <VG_SA>, where P means *past* (verb tense) and A means *active* (verb mood)).
2. **Detecting local predicates from syntactic and semantic rules.** These predicates specify semantic links between two or three consecutive chunks. Their detection is almost always triggerd by a annotated type. The other cases start from syntactic clues.
   For example, the predicate *rank_of(Arg1, Arg2)* specifies that *Arg2* is the rank of *Arg1*. That predicate triggers the <range_objet> tag, which marks chunks such as *"last", "the first two"* and so on. The predicate links such tags to the associated nominal phrase.
   There are about fifteen such relations. Some specify the object of the query, as in the example of Figure 5, where the relation *number_of* specifies which chunk is related to the *"how many"* question. Others expose a semantic link between two tags: for example, the relation *type_of* links the two tags of the following phrase:

```
<GN>
   <det> the </det>
    <topic><topic_cinema><pers_act><Acinema>
        director
    </Acinema></pers_act></topic_cinema></topic>
</GN>
<pers>
    Fritz Lang
</pers>
```

These local relations include an initial detection of coordinations *(and, or, etc.)*. These coordinations link chunks with identical tags. In the example of Figure 3, the scope of the coordination *and* has been detected from the presence of the same tag (<*pers*>), which marks the main chunks of the previous and following nominal phrases.
For very simple queries, for instance requesting a country's capital or a date of birth, these local relations are sufficient enough to express the whole semantic relation.

0	<Dneg> no it's not </Dneg>	1
	<PNG_on>	
1	<prep> about </prep>	2
2	<neg_info> <pers> Fritz Lang </pers> </neg_info>	3
	</PNG_on>	
3	<conjc> but </conjc>	4
	<PNG_on>	
4	<prep> about </prep>	5
5	<pers> Freud </pers>	6
	</PNG_on>	

Coordinations = and(not(1-2),4-5)

Fig. 3. Example of coordination of chunks

3. **Detecting global predicates**.
   Global relations aim at modeling patterns of generic queries. Four predicates are currently defined and tested:
   – **geo_record** involves queries such as *"what is the highest mountain of Soudan?"*. The predicate has three arguments: the type of the record (*highest*), the object of the record (*mountain*) and its domain (*Soudan*). The question mark in an argument specifies an association to the user query (*mountain*). Figure 4 shows the analysis and the formula related to this query.
   – **record** is a generalization of the previous. Its detection is tied to the tag <stat_object_plus>, typing chunks which express a superlative.

0	<Qquel> what is the </Qquel>	1
1	<stat_objet_plus> highest </stat_objet_plus>	2
2	<loc> <montagne> mountain </montagne> </loc>	3
	<PNG_of>	
3	<prep> of </prep>	4
4	<loc><pays> Soudan </pays></loc>	5
5	</PNG_of>	6

Formula = geo_record(1-2, 2-3?, 3-5)

Fig. 4. Example of global relation

- **to_create** involves the creation of an artistic work or a concept, or the discovery of an object. The three arguments are: the creator, the created object and the circumstances.
  In the example in Figure 5, only the first two arguments are instantiated: the first one (*Fritz Lang*) is the creator. The second one (*how many movies*) is associated to the created object and to the user's query. The third argument is uninstantiated. In this example, the detected predicates are linked with the logical connective *and*.
- **to_receive** has four arguments: the first is the person who receives (if any), the second is the object which receives (if any), the third is the reward received and the last the circumstances. For example, for the query: *"what is the movie which won the palme d'or in Cannes in 2001"*, the first argument is uninstantied, the second is *movie*, the third is *"palme d'or"* and the last is the list {*Cannes, 2001*}.

4 Present and Future works

We have currently only implemented the detection of the described relations in user queries. For performance reasons, all these searches are based on exploration of n-tree structures, and are written in C. Figure 6 shows the precision and recall, related to the detection of global relations; they are calculated from about 2500 user utterances. The local relation *geopol* is related to requesting a country's capital.

While the described relations are very frequent in our corpus[3], the coverage is obviously insufficient. First we will have to expand the kind and number of detected relations. We will also have to apply this detection to the documents in order to test the contribution of the approach to the QA system.

Otherwise, while keeping simple generic predicates helps to unify the semantic relation between queries and documents, it is in no way sufficient. For

[3] The relations of figure 6 are detected about 650 times in 4948 user utterances of the corpus, including dialogue interaction utterances, where there are no semantic relations to be detected.

0	<Qdial> *I want to know* </Qdial>	1
1	<Qnombre> *How many* </Qnombre>	2
2	<topic><topic_cinema><Tcinema>	
	movies	
	</Tcinema></topic_cinema></topic>	3
3	<pers> *Fritz Lang* </pers>	4
	<VG_PA>	
4	<aux> *has* </aux>	5
5	<action> *written* </action>	6
	</VG_PA>	
6	<conjc> *and*</conjc>	7
7	<action> *directed* </action>	8

Detected relations:

$$\mathsf{number\_of?}(2-3)\ \mathit{and}\ \mathsf{and}(4-5, 7-8)\ \mathit{and}\ \mathsf{to\_create}(3-4, 1-2?, \_)$$

Fig. 5. An example of detected semantic relations from a user utterance: *"I want to know how many movies Fraitz Lang has written and directed"*.

Relation	**Precision P**	**Recall R**	**F_measure 2RP/(R+P)**
geopol	99%	90%	94%
to_create	93%	62%	74%
geo_record and record	98%	87%	92%
to_receive	Insufficient data		

Fig. 6. Number of detections for global relations

instance, *Henry IV was murdered by Ravaillac in 1610* can be represented by the logical formula

$$kill(Ravaillac, Henry\ IV)\ and\ date(kill, 1610)$$

while the query *When did Henry IV die?* can be represented as

$$dead(Henry\ IV)\ and\ date(dead, ?).$$

Without knowledge of the world, the information does not unify with the question and no answer is found. So we plan to add such knowledge under the form of deduction rules as has been done for the very simple example given here:

$$\forall X \forall Y\ (kill(X, Y) \Rightarrow dead(Y))$$

Our project is quite ambitious, perhaps even a little too ambitious, but given that the QA system already works as-is, any additional semantic information we manage to add can only help the performance. As such, we will be able to directly evaluate the contributions of our approach and ideas.

References

[BUC] Buchholz S., Daelemans W.: SHAPAQA: Shallow Parsing for Question Answering on the World Wide Web. In Proceedings of RANLP 2001, (2001).

[CUI] Cui H., Sun R., Li K., Kan M-Y., Chua T-S.: Question Answering Passage Retrieval Using Dependency Relations. Proceedings of ACM SIGIR, Salvador - Brazil (2005) 400–407.

[GIR] Girju R., Badulescu A., Moldovan D.: Automatic Discovery of Part-Whole Relations. Computational Linguistics **32** (2006) 83–135.

[GRI] Grishman R.: The NYU System for MUC-6 or Where's the Syntax? Proceedings of the MUC-6 workshop, Washington (1995).

[HEA] Hearst M.: Automatic Acquisition of Hyponyms from Large Text Corpora. Proceedings of the Fourteenth International Conference on Computational Linguistics, Nantes, France, July. (1992).

[NAR] Narayanan S., Harabagiu S.: Question Answering based on Semantic Structures. International Conference on Computational Linguistics (COLING 2004), Geneva, Switzerland, August 22-29, (2004).

[PRA] Pradhan S., Ward W., Hacioglu K., Martin J., Jurafsky D.: Semantic Role Labeling Using Different Syntactic Views. Proceedings of the Association for Computational Linguistics 43rd annual meeting (ACL-2005), Ann Arbor, MI, June 25-30 (2005)