Answer type validation in question answering systems

Arnaud Grappy, Brigitte Grau

Answer type validation in question answering systems*

Arnaud Grappy
LIMSI/CNRS
B.P. 133 91403 Orsay Cedex
France
Arnaud.Grappy@limsi.fr

Brigitte Grau†
LIMSI/CNRS
B.P. 133 91403 Orsay Cedex
France
Brigitte.Grau@limsi.fr

ABSTRACT
In open domain question-answering systems, numerous questions wait for answers of an explicit type. For example, the question "Which president succeeded Jacques Chirac?" requires an instance of president as answer. The method we present in this article aims at verifying that an answer given by a system corresponds to the given type. This verification is done by combining criteria provided by different methods dedicated to verify the appropriateness between an answer and a type. The first types of criteria are statistical and compute the presence rate of both the answer and the type in documents, other criteria rely on named entity recognizers and the last criteria are based on the use of Wikipedia.

Keywords
question answering system, answer type, answer validation

1. INTRODUCTION
Questions answering (QA) systems look for the answer of a question in a large collection of documents. The question is in natural language (example: Which president succeeded Jacques Chirac?).
In a first step, QA systems select text passages. Then, in a second step, the answer is extracted from these passages, according to criteria issued from the question analysis.

A basic strategy in all QA systems consists in determining the expected answer type which is then related to named entities (NE) present in the selected passages. The ability of QA systems to recognize a great amount of answer types is related to their powerfulness for extracting right answers ([6],[9],[15]). However, it is not possible to predict all possible expected type and to recognize instances of all of them in texts: only types corresponding to NE are so recognized (the NE classes correspond classically to those defined in MUC [5], plus other ones specially used in QA systems like film titles, book titles...). So, QA systems have to develop answer type validation in a dynamic way for filtering or validating answers. This work takes place in this last paradigm.

Answer validation aims at verifying that an answer given by a QA system to a question is valid: the answer is correct and justified by the associated text fragment. For example the question "Quel président succèda à Jacques Chirac ? (Which president succeeded Jacques Chirac ?)" waits for an answer that is a kind of president and more generally a kind of person. Type validation will allow to discard the candidate answer "Michel Rocard" extracted from the text passage "Michel Rocard succède à Jacques Chirac au poste de Premier Ministre" (Michel Rocard succeeded Jacques Chirac as Prime Minister) because it is recognized as a person. So, a module able to verify all kinds of type made explicit in questions could eliminate a lot of bad answers given by QA systems, as we will see section 8.

The validity of an answer type will be checked on all questions that expect an answer that is an instance or an hyponym of an entity type corresponding to physical objects, or to the classical named entity types. Thus only factoid questions are considered, not definition questions or why/how questions.

Type validation cannot only be based on the passage containing the candidate answer. In a study made on textual passages proposed by QA systems that have participated to the French EQueR campaign [4], it goes out that passages in which only one question word was missing (111 passages), 27% of missing words were the type given by the question.

The approach we present in this paper is based on the use of different methods that are each dedicated to verify the answer type and are chosen either for their high accuracy degree or their high recall degree, in order to dispose of complementary criteria. Each of them returns a value (the type is validated or not), and they are all combined by a learning method to obtain a final decision. The first kinds of method are statistical methods based on the co-occurrence of an answer and its type in a document collection. The second ones use NE recognizing module to reject bad answers or to provide a knowledge base. A third type of methods makes use of Wikipedia [1], the famous encyclopaedia, either by using

*This work has been partially financed by OSEO under the Quæro program.

†and ENSIIE : Ecole Nationale Supérieure d'Informatique pour l'Industrie et l'Entreprise

[1]Wikipedia: http://fr.wikipedia.org/wiki/Accueil

specific rules encoding the relation between an answer and its type or by looking for the answer type in the answer Wikipedia page. Finally, the learning method for combining these features relies on a decision tree. This work was done on the French language.

In this paper, section 2 presents the state of the art. After describing how answer types are determined section 3, section 4 describes how we apply NER systems, section 5 shows the use we make of Wikipedia and section 6 the statistical approach on two corpora. Then we explain section 7 how the different criteria given by the preceding methods are combined in order to provide a final decision before evaluating our results (section 8).

2. RELATED WORK

All question answering systems use NE. NE are textual objects (words or phrases) that can be categorized in classes. Four classes are usually used: person, location, organization and date. Question answering systems often define more classes. Sekine et al. [15] use 200 classes, Hovy and Al.[9] 122 and Harabagiu, et Al. [6] use WordNet synsets.

Named entities allow to select answers whose type fits the NE type expected by the question, if it can be deduced. In Answer Validation, NE verification can be used to reject answers as the answer "Paris", recognized as a LOCATION, to the question " Quel président succèda à Jacques Chirac ? (Which president succeeded Jacques Chirac?)" that waits for a kind of PERSON. However, this verification is limited to the set of predefined types and cannot be used to know if an answer corresponds to the expected type for all possible types.

Schlobach et Al. [14] present a method of type verification in the case where the answer is a location. This specificity allows using knowledge bases, as an ontology and WordNet. Information provided by these bases is combined by a learning method with statistic criteria, based in particular on the co-occurrences of the answer and the answer type. The evaluation is made by calculating the gain in a QA system.

This work was extended to the open domain [13]. The methods are similar: a combination of scoring methods with WordNet information. The WordNet method looks for a path between the answer and the answer type in WordNet. For their scoring scheme, they assess the likelihood that an answer is of a semantic type by estimating the correlation of the answer and the type in Web documents and propose several correlation measures. Another measure is also used that looks for "ANSWER is a TYPE" in the documents. The evaluation shows that this verification increases the MRR measure of a QA system (20 %).

Type validation is also related to the work presented in [8]. This article presents a method obtaining hypernym/hyponym pairs that uses semantic patterns like "ANSWER is a TYPE". At first, the method tries to instantiate the patterns in a document corpus which provides a first hypernym/hyponym set. New patterns are then collected by searching the elements of these sets. Finally new pairs are collected thanks to these patterns. However this method cannot be used in our case because it is not possible to foreseen or collect all possible pairs (answer/type).

3. ANSWER TYPE

The expected answer type is determined by the question analysis module of FRASQUES [3], a shallow parser that makes use of syntactic and semantic criteria, the question word and the syntactic form of the question. It provides two different kinds of answer types:

- **the specific type** is the type explicit in the question. For example, the specific type of the question "Quel acteur a joué dans Danse avec les Loups ? (Which actor played in Dance with Wolves ?)", is "acteur (actor)";
- **the NE type** that corresponds to the NE expected by the question. In the above question, the NE type is PERSON.

Different cases are thus possible:

- the specific type is equal to the NE type;
- the specific type is more accurate than the NE type;
- the specific type does not correspond to a NE type.

In order to train the scoring modules or to evaluate NE recognition modules, we built a learning corpus from the results provided by the QA campaign EQueR on French[2] in which involved systems have to return answers (exact answer and passages) to 500 questions. Among them, the 198 questions that explicit a specific type are kept to build the corpus. The corpus contains thus 98 different types as many questions expect the same type of answer.

Some types like "lieu (location)" are general, others like "bisquine (small boat)" are very specific. This occurs either when a NE type is expected "parc (park)" or not "traitement (treatment)". For example, some types are "movie, chief, president, island, ambassador, event, newspaper, composer, faculty, horse, biscuit".

Answers are given by systems involved in the campaign that could return five answers and textual passages to each question. To obtain a balanced corpus, we reduced the number of answers that were not of the expected type. Finally 2720 pairs answer/expected type form the learning basis with as many positive answers, in which answers are of the expected type (1360), as negative answers.

The rest of this article presents the different type checking methods: use of NE, use of Wikipedia and statistical measures.

4. USE OF NAMED ENTITY RECOGNITION SYSTEMS

4.1 Answer filtering

First, recognized NEs make possible to reject those answers whose NE type does not correspond to the expected NE type. For example, the question "En quelle année eut lieu la

révolution russe ? (In which year did the Russian revolution took place?)" waits for a date. This module will reject the answer "Alexandre Issaievitch Soljenitsyne", tagged as a person.

As the FRASQUES analysis module is applied to extract NE expected type, we also apply the NER module of this QA system to analyse the passages and the answers of the corpus and find NE. This module recognizes twenty NE types organized following the four classical types (person, location, organization, date). For example, location type contains "city" and "country" types. Numeric expressions (length, speed, etc.) are also recognized. The general type ProperName is added to tag proper names that are not recognized as instances of known NE types.

Four cases are possible:

- The question does not wait for a NE type. For example, "Quel oiseau est le plus rapide d'Afrique ? (What bird is the fastest in Africa?)". This module cannot provide any information and the value UNKNOWN is returned.
- The question waits for a NE type and the answer is not a tagged NE. The answer is seen as bad and the value NO is returned.
- The question waits for a NE type and the NE type of the answer cannot be assimilated to it. For example the number "300" for a question that requires a person. The answer does not have the right type and the value NO is returned.
- the expected NE type and the answer NE type are consistent: same type or same category, as location or ProperName for the expected type country. The answer is almost of the expected type and the value YES is returned.
  This kind of verification only gives a general idea of the answer validity. For example, "Michel Rocard" is considered as a "president" by this module.

Table 1 presents the results obtained by this method. It presents the number of answers of the expected type and the number of answers not of the type for each given value: the module concludes that the answer is of the type (YES), it concludes that is not the case (NO) and it cannot provide any information (UNKNOWN).

given value	#A. of the type	#A. not of the type
YES (1411)	**885 (63 %)**	526 (37 %)
NO (457)	132 (29 %)	**325 (71 %)**
UNKNOWN(852)	344(40 %)	508 (60 %)

Table 1: Results of the NE method

We can see that when the answer is seen as wrong, the answer is generally not of the expected type (71 %). When the value YES is returned, it is difficult to know if the answer is of the type or not (only 63% answers are of the specific type). Table 2 presents another evaluation. It shows that recall is low, due to the large number of values UNKNOWN.

Accuracy	recall	f-measure
0.65	0.45	0,53

Table 2: Evaluation of the filtering

4.2 Validation of answers

Recognized NE can also be used as a knowledge base. NE recognized in a large corpus allow to build list of words corresponding to specific types. The method searches the answer in the expected type list and if the answer is found then it is probably an instance of the expected type.

This kind of validation could be efficient if the number of classes of NE is large enough. Thus, we retain the NE lists collected by the RITEL [12] NE module that is able to recognize 274 types that can be as specific as "religion" or "fleuve (river)". Nevertheless, the NE type number is limited so all cases cannot be covered. Three cases are possible:

- There is no correlation between the expected type and one of the known types. The module cannot know if the answer is of the specific type or not and the value UNKNOWN is returned.
- The answer belongs to the type instance list. So it is probably correct and the value YES is returned.
- The answer is not in the type instance list. So it is seen as wrong and the value NO is returned.

Tables 3 and 4 give an evaluation of the method. We can see that when the expected type corresponds to a known type, the given value is often right (accuracy 0.75). The low recall (0.32) can be explain by the very high number of UNKNOWN values (57 %).

given value	#A. of the type	#A. not of the type
YES (656)	**506 (77 %)**	150 (23 %)
NO (515)	138 (27 %)	**377 (73 %)**
UNKNOWN (1549)	716 (46 %)	833 (54 %)

Table 3: results for NE lists

Accuracy	Recall	F-measure
0,75	0,32	0,45

Table 4: Assessment of NE validation

5. USING WIKIPEDIA

5.1 Search in specific pages

This method is based on the idea that Wikipedia[2] is an encyclopaedia in which each of its pages defines the elements constituting its title. So, we assess that if the expected type is found in the answer page, the title is probably an instance of the type.

The method looks for the type in Wikipedia pages whose title contains the answer. Three cases are possible:

[2]Wikipedia: http://fr.wikipedia.org

- No page title contains the answer. Nothing can be deduced and the value UNKNOWN is returned.
- The page corresponding to the answer contains the type. The answer is probably of the expected type and the value YES is returned.
- The page does not contain the type. The answer is not of the expected type and the value NO is returned.

Tables 5 and 6 present the results obtained by the method.

given value	#A. of the type	#A. not of the type
YES (661)	**491 (74 %)**	170 (26 %)
NO (589)	228 (39 %)	**361 (61 %)**
UNKNOWN (1470)	641 (43 %)	829 (57 %)

Table 5: Method results

Accuracy	Recall	F-measure
0,68	0,32	0.43

Table 6: Method evaluation

The first one shows that the method can be reliable when YES is returned (answer is of the type in 74% cases) but not in cases where the value NO is returned (only 61%). This can be explained by some types that are replaced by one of their synonyms in the page corresponding to the answer. The tables show that many answers do not have a Wikipedia page devoted. Thus, this validation method is rather reliable but it cannot cover all the possible cases.

5.2 Using extraction patterns

The next feature also takes advantage of Wikipedia, with the examination of all the pages. Some sentence structures allow to make explicit that the answer is of the expected type, as with such a form: **ANSWER is a TYPE**. Five sentence patterns, obtained by studying a corpus, have been conceived:

- **ANSWER être(be) DET**[3] **TYPE** (Nicolas Sarkozy est le (is the) président (president)).
- **TYPE ANSWER** (president Nicolas Sarkozy)
- **ANSWER, DET TYPE** (Nicolas Sarkozy, the president)
- **ANSWER (DET TYPE** (Nicolas Sarkozy (the president))
- **ANSWER: DET TYPE** (Nicolas Sarkozy: the president)

To know if an answer is of a type, for each answer/specific type, each pattern is instantiated, while TYPE takes the expected type value and ANSWER the answer value. Then, a query is built from these phrases and given to the search engine Lucene [7]. It searches one of the phrases in the Wikipedia pages. If a page is found, then the answer is considered to be of the type (the value YES is returned) otherwise it is not (value NO).

Tables 7 and 8 present the method results. They show that the method is reliable when the value YES is returned (73 % of right results). We can also see that all pair values are evaluated. By consequence, accuracy and recall are equal.

given value	#A. of the type	#A. not of the type
YES (974)	**713 (73 %)**	261 (26 %)
NO (1746)	647 (37 %)	**1099 (63 %)**

Table 7: Method results

Accuracy	Recall	F-measure
0,66	0,66	0.66

Table 8: Method evaluation

6. STATISTICAL MEASURES

The last features are statistical and informed by type and answer cooccurrences in a document set, whatever are the document or the relation between answer and type. The feature predicts that when a type and an answer are often found together in documents, then they are probably connected.

To rely answer and type occurrence frequencies to the type validity, a first learning method was tested, as in [14] and [13]. Its criteria are:

- **Occurrence rate**: the rate between the number of documents containing answer plus type and the number of documents containing the only type or the only answer. It measures cases where the answer is often present along with the type.
- **PMI** (Pointwise Mutual Information): the rate between the frequency of cooccurrences of answer and type and the product of answer occurrence numbers by type occurrence numbers.
  $PMI = \frac{Frequency(answer+expected\ type)}{Frequency(answer)*Frequency(expected\ type)}$
- **Occurrence Frequency**: type, answer and both element occurrence frequency. Those features complement the preceding one because they can distinguish cases of answer or type very rarely present in documents from cases where they often appear. Those different cases obtain different PMI values. For example, if an answer rarely appears in the data and the type often appears then the PMI is low although the answer often occurs with its type. These measures solve this problem.

Scores computed by these methods are given to a classifier (method bagging) that combines decision trees (cf. section 7), provided by the WEKA system[4].

Scores are computed on two collections: Wikipedia and a subset of the newspaper "Le Monde" from years 1992 to 2000.

[3]Determinant

[4]WEKA: http://sourceforge.net/projects/weka/

This last corpus corresponds to the corpus the answers are coming from. The number of occurrences of answers are then different in these two corpora, and it is worth comparing the scores that are thus computed.

Section 8 presents the evaluation of this method.

7. COMBINATION OF FEATURES

After creating features, the last step consists in combining them with a learning method given by WEKA software. The learning method is bagging which combines five decision trees.

Decision trees regroup cases with similarities. They look for the best feature for dividing the data, i.e. the one that makes the fewest mistakes. Then data are divided following this feature. The step is made until the better classification is found.

Bagging method combines decision trees. Each of them gives, eventually, a different value. Results are combined by a voting method to obtain a final value. Five decision trees are used and each of them has the same influence. So the final value is the one obtained by the majority.

Used features are:

1. NE filtering,
2. NE validation,
3. Type in the answer Wikipedia page,
4. Syntactic rules in Wikipedia pages,
5. Scores calculated on Wikipedia pages:
   - the ratio between answer+type cooccurrence number and answer or type occurrence number,
   - type, answer and type+answer frequency,
   - PMI measure (Pointwise Mutual Information),
6. Scores calculated on "Le Monde" articles.

In a first test, learning and test bases are the same. 90 % of given values are correct (answer is or is not of the type).

8. EVALUATION

To evaluate the proposed approach, three evaluations are conducted:

- The evaluation of all the features on the test base;
- The evaluation of the combining method;
- The evaluation of the approach on results of question answering systems.

The test set is constituted from the AVE 2006 [11] campaign data for French. In this campaign, triples made of a question, a potential answer and a passage are given to the participants that have to decide if the answer is validated (correct and justified by the passage) or not. Answers and passages are provided by question answering systems that have participated to the QA track for French at CLEF. For example, we can find such a triple:

- Question : Quel pays l'Irak a-t-il envahi en 1990 ? (Which country invaded Koweït in 1990 ?)
- Answer : Koweït
- Passage : En 1990, l'Irak a envahi le Koweït. (In 1990, Irak invaded Koweït.)

The test set contains 1547 answer/specific type pairs and half of the answers are of the specific type. Pairs correspond to 90 questions and 47 different types.

8.1 Features evaluation

We first evaluated all features separately. Table 9 presents the results with Accuracy (A), Recall(R) and F-measure(F).

feature	A	R	F
1) NE filtering.	0.69	0.54	0.60
2) NE validation.	**0.80**	0.32	0.45
3)Wikipedia page of answer	0.72	0.46	0.57
4)Extraction patterns	0.70	**0.70**	**0.70**
5)Scores on Wikipedia	0,68	**0,68**	**0,68**
6)Scores on "Le Monde"	0,70	**0,70**	**0,70**

Table 9: Features individual results

We can see that feature results on the test set are similar to those obtained on the learning base.

The method that looks for the type in the Wikipedia page of the answer obtains better results on the test set particularly on the recall measure (0,46 vs 0,32). The data distribution can explain that. In the test set, there are more answers that are persons than in the learning basis so there are more Wikipedia pages whose title contains the answer.

The second point to notice concern the statistic methods. They could not be evaluated in the preceding section. The table shows that 68% given values are correct which confirms that this method is relevant. We can also see that the statistic method using "Le Monde" is slightly better than the method using Wikipedia (70% right values vs 68%).

8.2 Evaluation of the combination

For evaluating the results given by the combination method, we will compare them with a baseline. We choose the NE filtering method as baseline, while all QA systems use at least a NE detection to filter candidate answers.

Table 10 presents the global results.

Method	accuracy	recall	F-measure
NE	0.69	0.54	0.60
Combination	0.80	0.80	0.80

Table 10: Global results

The table shows that 80% data are correctly classified. This high value shows that the method is efficient. Method results are clearly higher than those obtained by NE filtering and overcome results obtained by all features separately.

A study distinguishing cases that expect a NE type to those that did not was conducted. The idea was to know if same phenomena occur in the two cases.

test basis	correctly classified answers
NE type (1205)	82 %
not NE type (342)	74 %

Table 11: Type verification depending on named entities

Table 11 shows that answers that are expected to be a NE are ranked higher than those that are not NE. It can be explained by the higher ratio of answers with an expected NE type (78%).

Table 12 presents the confusion matrix of this evaluation. It shows that results are similar whatever the given value is (YES or NO).

given value	#A. of the type	#A. not of the type
YES	**603 (80%)**	149(20 %)
NO	159(20 %)	**636(80 %)**

Table 12: Confusion matrix

Let us see now some results:

- Hosni Moubarak is correctly seen like a president
- Yasser Arafat is correctly seen not to be a president
- Krypton is correctly seen like a planet
- Bethleem is correctly classified as not a planet
- Infortunately, Barings is not seen like a "big bank". It's probably due to the adjective.
- Dow jones is seen, wrongly, to be a company. The two words are often present together in documents although there is no hyponym relation between them.

8.3 Question answering systems improvement

After evaluating the method by itself, this section is dedicated to measure the possibility of improvement of QA systems using this module. To do that, we used data from the campaign AVE 2006 in which answers are given by different QA systems. Only few of them are correct answers (20%). Table 13 presents the correlations between the type validity values and the answer validity.

We can see that when the NO value is given, the answer is very often not of the type (92%). However nothing can be deduced when the YES value is given. The number of bad answers shows that answers of a specific type remain to be most often wrong (66%).

Table 13 shows that the module is 8% error. These cases are present when the given value is NO but the answer is validated. When the given value is YES the high proportion of invalidated answers (66%) correspond to bad answers that are of the specific type, for example the answer François Mitterand to the question "Quel président succèda à Jacques Chirac ? (Which president succeeded Jacques Chirac ?)". A filter that rejects answers not of the type decreases highly the proportion of answers not of the type (from 80 % to 66 %). However it decreases also the number of valid answer answers from 289 (total of valid answers) to 233 and rejects 19% of valid answers. These results show that this method has to be improved to be used as a filter.

given value	#correct answer	#uncorrect answer
YES (698)	**233 (34 %)**	465 (66 %)
NO (759)	56 (8 %)	**703 (92 %)**
Total	289 (20%)	1168 (80 %)

Table 13: Correlation between type validity and answer validity

AVE 2006 evaluation is made using only the given value YES. Systems participating to this task use a lot of verification criteria like common words in the text and in the question, their density, edition distance at the word level between the question and the passage... Lot of systems are relying on a type checking coming from the decision of a NER module. The best system on French, MLENT [10], shows a f-measure of 0.57. It uses a combination of a lot of features representing lexical, syntactic and semantic criteria. Our module leads to a f-measure of 0.48. This is due to the high number of incorrect answers that are of the expected type (when the value YES is given). The score shows that if this verification cannot lead alone to assess answer validity, it could improve other systems.

9. CONCLUSION AND FUTURE WORK

This article presents a method checking that an answer is of the specific type expected by the question. This method is based on a learning approach that makes use of different features: Named Entities checking, statistic measures based on occurrence numbers in corpus and the exploitation of the Wikipedia encyclopaedia. The method obtains very good results which show its effectiveness. It can be used to improve question answering system by checking all returned answers. However, it cannot be used alone to select the good answer.

In a previous work, [1], we studied answer validity by a learning method that incorporates a simple type validation. The next step will be to complete this system with this type validation method.

This work will also find its place in an answer validation module that decomposes the question in the different kinds of information to check. For example, for validating the answer "Pierre Béregovoy" to the question "Quel ministre se suicida en 1993 ? (Which minister committed suicide in 1993?)", it requires to show that the answer is a minister, that he committed suicide and that the action takes place in 1993. Our type validation method fits the type validation checking requirement.

10. REFERENCES

[1] A. Grappy, A.-L. Ligozat, and B. Grau. Evaluation de la réponse d'un système de question-réponse et de sa justification. In *CORIA*, 2008.

[2] B. Grau. Equer, une campagne d'évaluation des systèmes de question/réponse. *Journée Technolangue/Technovision (ASTI'2005),*, 2005.

[3] B. Grau, G. Illouz, L. Monceaux, P. Paroubek, O. Pons, I. Robba, and A. Vilnat. Frasques, le système du groupe lir, limsi. In *Atelier EQueR, Conférence (TALN'05)*, 2005.

[4] B. Grau, A. Vilnat, and C. Ayache. *L'évaluation des technologies de traitement de la langue: les campagnes Technolangue*, chapter 6, évaluation de systèmes de question-réponse. Traité IC2, série Cognition et traitement de l'information. Lavoisier, 2008.

[5] R. Grishman and B. Sundheim. Design of the muc-6 evaluation. In *MUC*, pages 1–11, 1995.

[6] S. M. Harabagiu, M. A. Paşca, and S. J. Maiorano. Experiments with open-domain textual question answering. In *Proceedings of the 18th conference on Computational linguistics*, Morristown, NJ, USA, 2000. Association for Computational Linguistics.

[7] E. Hatcher and O. Gospodnetic. *Lucene in Action (In Action series)*. Manning Publications Co., Greenwich, CT, USA, 2004.

[8] M. A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *In Proceedings of the 14th International Conference on Computational Linguistics*, 1992.

[9] E. Hovy, L. Gerber, U. Hermjakob, C.-Y. Lin, and D. Ravichandran. Toward semantics-based answer pinpointing. In *HLT '01: Proceedings of the first international conference on Human language technology research*, Morristown, NJ, USA, 2001. Association for Computational Linguistics.

[10] Z. Kozareva, S. vasquez, and A. Montoyo. Adaptation of a machine-learning textual entailmentsystem to a multilingual answer validation exercise. In *CLEF*, 2006.

[11] A. Penas, A. Rodrigo, V. Sama, and F. Verdejo. Overview of the answer validation exercise 2006. In *CLEF*, 2006.

[12] S. Rosset, O. Galibert, and A. Max. Interaction et recherche d'information : le projet ritel. *Traitement Automatique des Langues*, 2005.

[13] S. Schlobach, D. Ahn, M. de Rijke, and V. Jijkoun. Data-driven type checking in open domain question answering. *J. of Applied Logic*, 5(1), 2007.

[14] S. Schlobach, M. Olsthoorn, and M. D. Rijke. Type checking in open-domain question answering. In *In Proceedings of European Conference on Artificial Intelligence*. IOS Press, 2004.

[15] S. Sekine, K. Sudo, and C. Nobata. Extended named entity hierarchy. In M. G. Rodríguez and C. P. S. Araujo, editors, *Proceedings of* 3^{rd} *International Conference on Language Resources and Evaluation (LREC'02)*, pages 1818–1824, Canary Islands, Spain, May 2002.