# Adaptation of LIMSI's QALC for QA4MRE

Brigitte Grau[1,3], Van-Minh Pho[1,2], Anne-Laure Ligozat[1,3], Asma Ben Abacha[1], Pierre Zweigenbaum[1], and Faisal Chowdhury[1,4]

LIMSI-CNRS, rue John von Neumann, 91403 Orsay cedex, France,
Université Paris-Sud, 91400 Orsay, France
ENSIIE, 1 square de la résistance, 91000 Evry, France
FBK, Via S.Croce 77, 38122 Trento, Italy
firstname.lastname@limsi.fr

**Abstract.** In this paper, we present LIMSI participation to one of the pilot tasks of QA4MRE at CLEF 2012: Machine Reading of Biomedical Texts about Alzheimer. For this exercise, we adapted an existing question answering (QA) system, QALC, by searching answers in the reading document. This basic version was used for the evaluation and obtains 0.2, which was increased to 0.325 after basic corrections. We developed then different methods for choosing an answer, based on the expected answer type and the comparison between question plus answer rewritten to form hypothesis compared with candidates sentences. We also conducted studies on relation extraction by using an existing system. The last version of our system obtains 0.375.

**Keywords:** question answering

## 1 Introduction

In this paper, we present LIMSI participation to one of the pilot tasks of QA4MRE at CLEF 2012: Machine Reading of Biomedical Texts about Alzheimer. The objective was to select the correct answer from a list of five possible answers, according to a corresponding document, which was a biomedical text about Alzheimer disease. For this exercise, we adapted an existing question answering (QA) system, QALC [1]. We selected candidate sentences of the document of the reading test, depending on the question and answers terms present in the sentence, and their distance, as it was done in QALC. In order to improve answer selection, we enhanced our lexicons for variant recognition, and used an existing relation extraction module to detect the semantic relations between entities. We also added two criteria for selecting answers: verification of the expected answer type, and similarity measure between question + answer and candidate sentence, according to shallow or syntactic measures.

We used the background collection in order to collect lexicons and to verify answer expected types. We built a list of terms associated with their UMLS concept, and a list of definitions extracted with pattern on the annotated collection.

In the following, we will first present the adaptation of our QA system, and after the new modules we developed. We will then present results. As our results

were very different on the development set and on the test set, we randomly selected 10 questions among the 4 reading tests in order to study our errors and to augment the training set. Our official results were 8 right answers on 40 questions. After the new developments, we gained 5 correct answers on the remaining 30 questions.

## 2 Methods

### 2.1 Adaptation of the existing Question Answering system QALC

As we already disposed of a question answering system for English [1], we used it as a basis for this task. The architecture of this adapted system, named QALC4mre, is presented in Figure 1. We reused existing modules concerning variant recognition of terms and adapted the sentence weighting scheme for passage selection, in order to integrate the presence of an answer.

We only look for answers in the document of the reading test, in full text, and passages are made of one sentence. We normalized Greek letters, in order to standardize document, questions and answers.

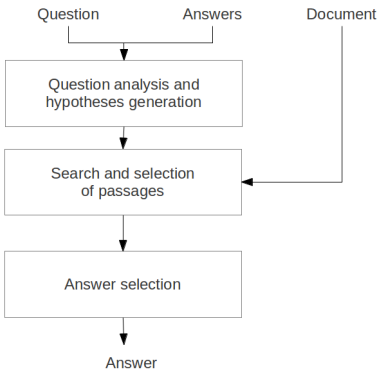


**Fig. 1.** Architecture of QALC4mre

We now detail each step of the answering process.

### 2.2 Question analysis

The objectives of the question analysis module are the following:

– determine the expected answer type: for example *enzyme* in the question *Which enzyme is responsible for the transformation of testosterone into estrogen?*;
– rewrite the question to form a hypothesis: for example, from the previous question and the candidate answer *aromatase*, the hypothesis should be *aromatase is responsible for the transformation of testosterone into estrogen.*

This module was developed for QA4MRE, as types of questions and answers are very different from factual question in open domain. Determination of the expected answer type is based on the question syntactic tree. The question is parsed with the Biomodel tool [2]. Tregex and Tsurgeon [3] are used to determine the expected type of the answer according to its position in the parse tree: we basically choose a common or proper noun, that is a son of the interrogative.

Then, the question is rewritten in the assertive form thanks to Tsurgeon rules, and the answer replaces the interrogative, leading to generate five sentences which we will name hypothesis. These hypothesis will be compared to selected sentences, in order to contribute to the choice of the answer

All rules were developed on a separate corpus of nearly 300 medical questions extracted from the Journal of Family Practice[1]. On this set of questions, 94% expected types are correct.

### 2.3 Sentence selection

Sentence selection is based on recognition of the hypothesis terms in the sentences of the reading document: question terms and answer terms.

As in QAlC, we look for multiword terms and monoword terms, either their exact formulation or their variants, by using Fastr [4]. Fastr analyses these sentences and recognizes morphological, syntactical and semantic variants of terms by applying rewriting rules on tagged documents with POS tags. For example, if the hypothesis contains the word *association*, the word *associated* will be recognized as a morphological variant. Besides uni-terms, multiword term variants are also recognized:*aged male mice* is recognized as a variant for *old mouse*, *patients with AD* for *AD patient* or *regulates IDE expression* for *expression regulation.* We noted that in the development test, some verb variants were missing, such as *convert/transformation.* In order to add such variants to our lexicons, we added the variants contained in the FrameNet lexical units to our lexicon. This list contains 673 entries, with several variants for each entry (for example the name *modification* has the following variants: the verb *change*, the verb *convert*...). It contains some useful variants for our task, such as *convert/transformation*, but also antonyms, such as *successful/failed.* Adding these variants to our lexicons enabled us to recognize more terms, but did not improve the overall performance of the system.

Fastr was applied on full text tagged documents and on normalized document with terms replaced by their UMLS concept, named CUI. A list of the

[1] www.jfponline.com

different CUIs associated to terms was extracted from the annotated background collection in that purpose.

**Sentence weighting** Each recognized term is attributed a weight, corresponding to a combination of different criteria. The criteria that we retained are those of QALC and use the following features retrieved within the candidate sentence:

- question words, weighted by their specificity degree,
- variants of question words,
- exact words of the question,
- mutual closeness of question words.

The main item is the specificity degree of the terms. This value depends on the inverse of the term relative frequency within a large corpus of newspapers. As the task domain is Alzheimer disease, there are specific terms which do not belong to the corpus, thus they receive a weight of 1, i.e. the max value. First we compute a basic weight of the sentence based on the presence of question words within the sentence, and then we add weights from the other criteria. The computation of the basic weight of a sentence is made from lemmas (or from words if the word is unknown for the tagger), and their specificity degree. Some words are not taken into account, i.e. determinants or prepositions, transparent nouns, and auxiliary verbs. A transparent noun is a noun whose complement is semantically more relevant that the noun itself. For instance, the word kind is transparent in a question as What kind of gliar cell ... , and cell is the semantically relevant noun. We made an a priori list of such words.

Thus, the basic weight of a sentence is given by:

BasicWeight = (dr1 + ... + dri + ... + drm) / (dq1 + ... + drj + ... + dqn)

with:

dri: dri is a question term found in the sentence. If dri is a mono-term, the specificity degree of its lemma, 0.1 otherwise,

dqj: drj is a question term. If dqj is a mono-term, the specificity degree of the lemma, 0.1 otherwise,

m: number of lemmas found in the sentence,

n: number of lemmas in the question.

Each lemma can be taken into account each time it occurs in the same sentence or only once. If a word from the question is not found in the sentence, but a variant of it, half of the specificity degree of the word is added to the basic weight of the sentence. As the elementary weights belong to [0-1], the basic weight maximum is close to one. We bring it to 1000 for convenience. We subsequently add an additional weight to this basic weight for each additional criterion that is satisfied. Each additional weight cannot be higher than about 10% of the basic weight. The criterion of mutual closeness of question words aims at representing the fact that several words are used in the same way in the question and in the sentence. Thus, it is computed between single terms arranged in pairs in the sentence. Each pair which is separated by maximum a

significant word receives a weight of 0.02. The last criterion represent the ratio of lemmas found in the sentence without variation.

Thus, the final weight W of a sentence S is:

W(S)= BasicWeight*1000 + MutualCloseness*1000 + ExactLemmas*100

Candidate answers are also searched in the sentences, and they are weighted by the following scheme:

W(a)= BasicWeight*1000 + ExactLemmas*100

**Relation extraction** Semantic relation extraction consists in determining the relations linking two given named entities. This task uses a predefined context containing the two entities (source and target of the relation) which may be a sentence, a paragraph or a whole document. This textual context can be exploited with diverse techniques (morpho-syntactic analysis, dependency relations, synonymy, etc.) for the identification of semantic relations.

In the biomedical domain, several approaches tackled relation extraction between (bio)medical entities such as the treatment relation between a treatment and a disease or protein-protein interactions. However, relation extraction in the context of precise information retrieval systems such as question-answering systems is not as widely covered in the literature.

Two main methods could be described for relation extraction:

- Pattern-based / keyword-based methods which use a list of patterns or keywords to identify the semantic relation. Table 1 presents some examples of keywords associated to the relation "inhibit".
- Machine learning methods which allow to build automated classifiers with annotated corpora. This second category of approaches is the most scalable when a sufficient number of training examples is available for the targeted relation.

| | |
|---|---|
| Wordnet | inhibit, inhibitor, inhibition, limit, block, decrease |
| Xiao and Rosner[5] | suppress, restrict, reduce, prevent, restrain |

**Table 1.** Keywords examples for the "inhibit" relation

In our current work, we used a machine learning method and trained a classifier for Protein-Protein Interaction (PPI) and the Regulatory Relation (RR). The annotations of biomedical entities were provided by the application of two systems on the reference corpora: GDep parser and ABNER tagger. Chunks and named entities are represented in the BIO format (B for Begin, I for Inside, and O for Outside). Five semantic classes of medical entities were annotated: DNA, RNA, cell_line, cell_type, and PROTEIN. We list three example annotations of the reference corpus showing, respectively, an entity recognized by the first tool, by the second tool, then by both tools:

– glycoprotein glycoprotein I-NP NN B-protein 4 SUB B-C0017968 O
– KEYWORD KEYWORD B-NP NN O 0 ROOT O B-PROTEIN
– APP APP B-NP NN B-protein 14 PMOD B-C0085151 B-PROTEIN

We used the following annotated corpora for the extraction of PPI and RR:

1. **PPI**: 5 PPI benchmark corpora
   – AIMed[2], BioInfer[3], HPRD50[4], IEPA[5], LLL[6],
2. **RR**: BioNLP-ST 2011[7]. Other annotated corpora exist for this relation[8].

### 2.4 Answer selection

#### Answer type validation

*Corpus validation* Some questions contain the expected type of the answer: for the question *What experimental technique was used specifically to purify the -secretase complex?*, the answer must be an *experimental technique*. This information can be found in corpora, for example by searching for existing hyponymy patterns between the answer and the expected answer type. We used the annotated background collection for this purpose. We extracted all dependency relations of the *NMOD* type (noun modifier) between two nouns, since this was the most common way the hyponymy was expressed: for example, the noun phrase *affinity chromatography technique* can be found in the background collection, and validates *affinity chromatography* as a possible answer to the previous question. Then, for each possible answer, we searched for the presence of the head of the answer and the head of the expected answer type in the previous list of relations. We attribute a score to the instantiated patterns:

– 3 if all the answer belong to the extracted definition
– 2 if the definition contains the head word of the answer, computed as its last word if it is a noun,
– 1 if there are other words than the head word.

In order to enhance the recall of this method, and because other kinds of extraction patterns are difficult to conceive, we also look for cooccurrences of question words with the expected type in the full text collection, by searching short passages with Lucene, and counting cooccurrences in excerpts of sentences without punctuation marks [9] which could indicate separate phrases. The score given by this method is the number of extracted cooccurrences.

[2] ftp://ftp.cs.utexas.edu/pub/mooney/bio-data/
[3] http://mars.cs.utu.fi/BioInfer/
[4] http://www2.bio.ifi.lmu.de/publications/RelEx/
[5] http://class.ee.iastate.edu/berleant/s/IEPA.htm
[6] http://genome.jouy.inra.fr/texte/LLLchallenge/
[7] https://sites.google.com/site/bionlpst/home/bacteria-gene-interactions
[8] e.g. (i) http://www.bork.embl.de/Docu/STRING-IE/ and (ii) http://maya.ccg.unam.mx/ccg-ie/
[9] Such as , or ; or ? or ()

*UMLS validation* Given the expected type of the answer and a candidate answer, we also check whether the former subsumes the latter in an external resource with a large coverage of the biomedical domain: the UMLS Metathesaurus [6]. We do this by first projecting them to UMLS concepts, then by testing concept pairs for subsumption.

Since the background collection was annotated with UMLS concepts (Concept Unique Identifiers, or CUIs), we collected all terms annotated with CUIs into a dictionary. We then used this dictionary to annotate the expected type of the answer and each candidate answer with one or more CUIs. In case of multiple CUIs, no disambiguation was attempted. After the official submission, we also added a CUI detection method based on exact match to any UMLS string.

The UMLS Metathesaurus includes more than a hundred source vocabularies covering various sub-domains of medicine. Most of these vocabularies have a hierarchical structure which is often based on the is-a relation, but can also mix it with part_of or other relations. We focused on the hierarchies of six large vocabularies included in the UMLS Metathesaurus: the Systematized Nomenclature of Medicine (SNOMED CT), the Gene Ontology, the National Cancer Institute thesaurus, the International Classification of Diseases (ICD9-CM and ICD10-CM), the MeSH thesaurus, and the Medical Drug Regulatory Activity thesaurus. We queried these hierarchies through the pathsToRoot method of the UMLS::Interface Perl module [7], which provides all ancestors of a given CUI. The subsumption test was considered successful for a candidate answer if the expected type of the answer was found among its ancestors in any of the six vocabularies. Version 2011AA of the UMLS was used for these tests.

**Similarity between hypothesis and sentence** For selecting the correct answer among the five candidates, we consider that it must be the one that produces an assertive form which will be the closest of the supporting sentence. Thus we compute the similarity between assertive forms of the answers, named hypothesis, and sentences selected by the system. We chose metrics used originally for the recognition of textual entailment. These metrics are based on two levels of textual representation: surface and syntactic forms.

The metric based on surface forms of the hypothesis and the sentence is TERp (Translation Edit Rate plus) [8], which computes the edit distance between hypothesis and sentence. In addition to compute the minimal number of word insertions, deletions and substitutions, TERp includes phrasal shifts. In our case, we do not compare the hypothesis with the whole sentence: we keep the part of the sentence containing words of the hypothesis.

Most of the metrics we used are based on dependency trees of the hypothesis and the sentence. Here is the list of these metrics :

- the ratio of common dependencies between hypothesis and sentence to the number of hypothesis dependencies. Two dependencies are common if the father node, the relation and the child node are the same. For example, the following dependencies are equal :
  - be SUB NEP

- be SUB mouse NMOD level PMOD NEP

– the tree edit distance between the sentence and the hypothesis. We compute the minimal cost of operations to transform the dependency tree of the sentence into the dependency tree of the hypothesis. For this, we implemented Zhang and Shasha's algorithm [9];
– the ratio of common subtrees between hypothesis and sentence and the subtrees of the hypothesis. For this, we compute the tree kernel between both utterances [10].

We also compute the common subtrees between the constituent trees of hypothesis and sentence.

## 3 Results

### 3.1 Official results

The system version used for the official evaluation was the adaptation of QALC, applied on the full text documents, tagged with TreeTagger [11] and on the documents where terms annotated with an UMLS concept in the background collection were replaced by their concept identifier. In this test, questions and answers were also annotated in the same way. The scoring scheme for sentences counted all the occurrences of a same term within them.

Two scoring measures were implemented for selecting an answer:

– `max_sentence`: selects the answer with the highest weight from the sentence which has the highest weight,
– `most_frequent`: selects the most frequent answer from the N most important sentences (N=5).

The different measures produced the same scores of 8 correct answers on the test set (40 questions), while they produced 5 or 6 answers on the development set (10 questions). For studying errors, we randomly chose 10 answers among the four reading tests. Errors come from:

– Non recognized variants: synonyms, abbreviations, as familial forms of Alzheimer's disease *vs* FAD, and collocations, as medical condition *vs* disease,
– Ambiguity among several answers inside the correct sentence,
– Problem in the normalisation process (characters not encoded in utf-8, remaining Greek letters).

We then realized new tests, on the remaining 30 questions, after correction of the normalization of documents, and by adding type verification and similarity measures.

### 3.2 Type checking

UMLS subsumption could be checked on the test set when UMLS concept identifiers (CUIs) were detected for both a candidate answer and the expected answer type. This happened for 17 {answer, expected answer type} pairs, and the subsumption test was positive for 3 of these. For instance, for question *Which hormone can control the expression of CLU isoforms?*, the expected answer type was *hormone* (CUI C0019932), and only one answer (androgen, CUI C0002844 or C0919646, which is the correct answer) had a CUI which was subsumed by C0019932.

Type verification by computing cooccurrences was launched on 15 passages. We integrated type verification in the background collection in QALC4mre by computing two weights for the answers:

- `cooc`: Number of occurrences * 10 + pattern score,
- `pat`: pattern score * 100 + occurrence number.

The score `cooc` gives priority to the robust method, while `pat` favors precision. These scores were used in place of the weight computed according to the found terms. We computed two measures, by considering only not null scores for both scores, `cooc0` and `pat0`, or not, `cooc` and `pat`. Results are shown in table 2.

Sentences were ordered by the same weighting scheme of the official evaluation, `sent_weight_all`, and by considering only once multiple question terms in the sentence, `sent_weight_1`. In place of tagging documents of the reading tests, we used their annotated version given by the organizers.

| | `sent_weight_all` # | `sent_weight_1` # | best accuracy (30 questions) |
|---|---|---|---|
| `weight` | 9 | 13 | 0.43 |
| `pat` | 8 | 12 | 0.40 |
| `pat0` | 7 | 8 | 0.26 |
| `cooc` | 8 | 11 | 0.36 |
| `cooc0` | 5 | 7 | 0.23 |
| `most_frequent` N=5 | 12 | 13 | 0.43 |

**Table 2.** Evaluation on the new test set made of 30 questions

The correction of the documents and the use of the annotated reading documents lead to improve the system which was used for the official evaluation: 9 and 12 answers on 30 questions, column `sent_weight_all`, lines `weight` and `most_frequent`. On the 10 questions removed, we only find 1 answer. Thus our new score is 0.325 on the initial test set with the evaluation version of QALC4mre corrected.

We can see that the best sentence weighting is obtained by counting only once question terms in sentences. It was the initial weighting scheme of QALC;

however, on the development set, it seemed better to account for all the occurrences, as sentences were often long. In the test set, as many question terms are not recognized, the waiting scheme `sent_weight_all` overweights sentences with multiple occurrences of few question words, bypassing sentences with more different terms.

The measure which selects most frequent answers in the top 5 sentences shows the better results on the test set. Concerning the type checking scores, we can see it is better to keep an answer even if one of the two scores is null, and giving the priority when definition patterns apply seems more reliable.

### 3.3 Relation extraction

We applied the Hyrex system [12] for relation extraction, which uses SVM-LIGHT-TK[10]. Hyrex is currently the state-of-the-art system for the extraction of PPI.

**First Approach.** In this first attempt, we consider that a sentence that contains the same relation expressed in the question is more likely to contain the required answer. The application of the Hyrex system allowed to retrieve PPI and RR relations in the annotated corpora. However, very few relations were retrieved from the questions (e.g. only one PPI relation was retrieved in the 40 questions). This is mainly due to the fact that not all biomedical entities were retrieved by GDep parser and ABNER tagger. For example, in the question "*With which particular protein does amyloid-beta interact?*" only *amyloid-beta* was recognized, and the implicit entity referred to by "particular protein" was not detected. This first results led us to think of an adaptation consisting in using the declarative forms associated to the questions for the detection of biomedical entities.

**Second Approach.** In a second attempt we used the declarative forms associated to the questions. In a first step, we used the answers provided for each question to associate 5 declarative sentences to each question. For example, the following declarative sentences were associated to the question "*Which hormone is able to inhibit the transcription of BACE1?*":

- PB1P-A is able to inhibit the transcription of BACE1 .
- APP is able to inhibit the transcription of BACE1 .
- Testosterone is able to inhibit the transcription of BACE1 .
- IDE is able to inhibit the transcription of BACE1 .
- NEP is able to inhibit the transcription of BACE1 .

In a second step, we launched relation extraction with the Hyrex system. We evaluated only 10 questions.

With relation extraction only, we retrieved answers for 4 questions of the initial 10. Only one of these answers is correct: "knock-out of BACE1 gene" the answer of the question "What experimental approach was successful to inhibit

[10] http://disi.unitn.it/moschitti/Tree-Kernel.htm

in vivo the production of amyloid beta?" (precision value of 25%). However, these results must be taken with caution as relations were retrieved for only 6 declarative sentences among the 50 declarative sentences associated to the targeted 10 questions. Thus, the number of extracted relations was not sufficiently important to integrate the relation extraction module in the final system. However, this approach could increase the weight of the answers when relations are extracted from them.

We think that the second approach is very interesting, in particular if the affirmative forms associated to the questions are well constructed. What remains to be improved is: (i) to target more relation types and (ii) to use other annotated corpora to increase recall and precision.

### 3.4 Similarity between hypothesis and sentences

Similarity between hypothesis and sentences allows to select the most likely answer, for each question of the corpus. We evaluate our system with each metric plus the weight between hypothesis and sentences. We use evaluation functions described in the introduction of this section. We use two formulas to select the correct answer:

- `max_sentence`: selects the hypothesis with the highest similarity from the sentences which have the highest weight.
- `max`: selects the hypothesis with the highest similarity score, regardless of sentence weights.

Table 3 gives the results of this evaluation. The best results bypass the best result given by the `most_frequent` function (43 % correct answers in table 2). Although computation of tree edit distance between hypothesis and sentences gives the best results (whathever the evaluation function), other measures provide correct answers not selected by tree edit distance. We can see that similarity based on syntactic criteria lead to the best scores and seem to be worth to develop. When evaluated on the 40 questions of the initial test set, the system finds 15 correct answers and obtain an accuracy of 0.375.

| Metric | `max_sentence` | `max` |
|---|---|---|
| TERp | 0.43 | 0.43 |
| Common dependencies | 0.43 | 0.40 |
| Common constituent subtrees | 0.40 | 0.30 |
| Common dependency subtrees | 0.37 | 0.33 |
| Tree edit distance | 0.43 | 0.47 |

**Table 3.** Evaluation of the similarity between hypothesis and sentences

There are several reasons to select a wrong answer:

- the sentence justifying the good answer has a weak weight;

– the most relevant sentences do not contain the good answer;
– a wrong hypothesis has a better similarity score than the correct one;
– hypothesis of the most relevant sentences have the same similarity score. In this case, the first hypothesis is chosen.

We have to find other criteria or other similarity measures in order to answer the third kind of problem. For the last case, a possibility could consist in making a fusion.

## 4 Conclusion

The system we developed for QA4MRE is an adaptation of a QA system in open domain. The scoring scheme of sentences reveals to be adapted in this new task. The system found 13 correct answers for 30 questions, after basic correction. We studied different methods for selecting the right answer among candidate sentences: verification of the expected answer type by semantic verification in the UMLS and corpus verification, similarity measures between each hypothesis and the candidate sentences, based on surface or syntactic features. We also studied relation extraction in order to improve answer and sentence selection. All of these methods show interesting results, and we have to study how to integrate them. However, an important remaining problem is that many variations of question terms are still not recognized. We will study an integration of different lexicons and methods for bypassing absence of knowledge, as searching for paraphases in corpus.

## References


1. de Chalendar, G., Dalmas, T., Elkateb-Gara, F., Ferret, O., Grau, B., Hurault-Plantet, M., Illouz, G., Monceaux, L., Robba, I., Vilnat, A.: The Question Answering System QALC at LIMSI, Experiments in Using Web and WordNet. In: TREC11 NIST SPECIAL PUBLICATION SP. (2002) 457–467
2. McClosky, D.: Any Domain Parsing: Automatic Domain Adaptation for Natural Language Parsing. PhD thesis, Brown University (2010)
3. Levy, R., Andrew, G.: Tregex and Tsurgeon: tools for querying and manipulating tree data structures. In ELDA, ed.: Proceedings Fifth international conference on Language Resources and Evaluation (LREC 2006), Genoa, Italy, ELDA (2006)
4. Jacquemin, C.: Syntagmatic and paradigmatic representations of term variation. In: Proceedings of the 37th annual meeting of ACL. (1999)
5. Xiao, C., Rsner, D.: Finding high-frequent synonyms of a domain-specific verb in english sub-language of medline abstracts using wordnet. In: GWC 2004 - Proceedings of the 2nd International Conference of the Global WordNet Association. (2004) 242–247
6. Bodenreider, O.: The Unified Medical Language System (UMLS): Integrating biomedical terminology. Nucleic Acids Research **32**(Database issue) (2004) D267–270

7. McInnes, B., Pedersen, T., Pakhomov, S.V.: UMLS-Interface and UMLS-Similarity : Open source software for measuring paths and semantic similarity. In: Proceedings of the American Medical Informatics Association (AMIA) Symposium, San Fransisco, CA (November 2009)
8. Snover, M., Madnani, N., Dorr, B., Schwartz, R.: Ter-plus: paraphrase, semantic, and alignment enhancements to translation edit rate. Machine Translation **23**(2) (2009) 117–127
9. Zhang, K., Shasha, D.: Simple fast algorithms for the editing distance between trees and related problems. SIAM J. Comput. **18**(6) (1989) 1245–1262
10. Wang, K., Ming, Z., Chua, T.: A syntactic tree matching approach to finding similar questions in community-based qa services. In: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval, ACM (2009) 187–194
11. Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In: Proceedings of the International Conference on New Methods in Language Processing. (1994) 44–49
12. Chowdhury, M.F.M., Lavelli, A.: Combining tree structures, flat features and patterns for biomedical relation extraction. In: EACL. (2012) 420–429