Evaluation of a Baseline Information Retrieval for a Polish Open-domain Question Answering System

Michał Marcińczuk and **Adam Radziszewski** and **Maciej Piasecki**
Dominik Piasecki and **Marcin Ptak**
Wrocław University of Technology
{michal.marcinczuk,adam.radziszewski,maciej.piasecki}@pwr.wroc.pl
{dominik.piasecki,mp.marcin.ptak}@gmail.com

Abstract

We report on our efforts aimed at building an Open Domain Question Answering system for Polish. Our contribution is twofold: we gathered a set of question–answer pairs from various Polish sources and we performed an empirical evaluation of two re-ranking methods. The gathered collection contains factoid, list, non-factoid and yes-no questions, which makes a challenging material for experiments. We show that using two re-ranking methods based on term proximity allows to obtain significant improvement on simple information retrieval baseline. The improvement is observed as finding more answer-bearing documents among the top n search results.

1 Background

Question Answering (QA) is an information retrieval task in which the user information need is expressed in terms of a natural language question. As this way of expressing information needs is very flexible, QA systems are mostly constructed as Open Domain QA systems (ODQA), not limited to any particular text collection or narrow domain (Paşca, 2003). An ODQA system can deliver an answer, as a text of a data record, but mostly it is required that it returns passages extracted from a collection of documents that are supposed to include an answer to the user's question. The goal of ODQA is to answer questions not restricted to any pre-defined domain (Paşca, 2003).

Most ODQA systems process user questions in four steps, cf (Paşca, 2003; Monz, 2003; Ferrucci, 2012): question analysis, document retrieval, document analysis and answer selection. In addition to this general scheme, we can distinguish several typical substeps or tasks: question classification (Lally et al., 2012), query selection and expansion (Paşca, 2003), passage retrieval and ranking (Paşca, 2003), candidate answer identification (Chu-Carroll et al., 2012), answer extraction and ranking (Gondek et al., 2012), etc., but the core is shared among systems.

There are only a few known works on ODQA (working on text collections) for Polish, e.g. Walas and Jassem (2011), Walas (2012), and two systems publicly available: `Hipisek.pl` and `KtoCo.pl`. The latter is a commercial system and little is known about its structure. *Hipisek* implements the ODQA blueprint described above (Walas and Jassem, 2011), but was focused on processing *yes-no* questions about time and location. The system depends on a dedicated rule-based parser (Walas and Jassem, 2011), and was extended with a knowledge base for spatial relations (Walas, 2012).

Our long-term goal is large-scale, broad-application ODQA with respect to different types of questions and documents indexed. We utilize the following architecture of QA system:

- **query analysis** — query processing by language tools and rules, and generation of the search query,
- **search engine** — fetches the N most relevant documents from a large collection of documents,
- **module for document ranking** — the set of documents returned by the search engine is re-ranked using medium time-consuming techniques. The M top documents are selected, where $M \ll N$,
- **module for extracting candidate answers** — the most time-consuming operations are performed on the reduced set of documents,
- **module for answer ranking** — the list of candidate answers with their context are

Proceedings of Recent Advances in Natural Language Processing, pages 428–435,
Hissar, Bulgaria, 7-13 September 2013.

ranked and best items are presented to user.

In this paper we focus on the first three elements of QA system with the special focus on search engine and document re-ranking.

2 QA dataset for Polish

Most works performed for English rely upon the TREC datasets (Voorhees, 2001). No such dataset was available for Polish, so we started construction of a set of question-answer pairs for Polish. We surveyed several possible ways of collecting questions and answers for Polish from the available resources.

Our first idea was to crawl Internet QA communities such as `zapytaj.onet.pl`, `pytki.pl`, `pytano.pl` (Polish counterparts of `ask.com`) and grab both questions and answers. We anticipated the need for substantial manual work needed to select and curate the data, but the actual scale of the problem was quite overwhelming: while it is already not easy to find suitable questions there, finding a number of suitable answers in reasonable time was practically infeasible. The main problem was that if the answers would serve as a testing material for a system based on document retrieval, the answers should mimic normal documents. The answers posted by users of such sites are usually very short and devoid of the necessary context to understand them — they make sense only when paired with the corresponding questions (those that make sense at all). The same problem turned out to apply for FAQ sites, even official ones. This way we faced the necessity to divide the process into two separate phases: gathering questions and then finding documents that provide answers to them.

2.1 Gathering questions

We developed simple guidelines that help to recognise acceptable questions. A question must have syntactic structure of a question (rather than a string of query terms), be simple (one question per a sentence), not requiring any additional context for its interpretation. We did not accept questions referring to the person asked (‘*What bands do you like?*’). Questions about opinions were discouraged unless could be conceived as addressed to a domain expert (e.g. ‘*Which wines go well with fish?*’). We did not exclude questions which were vulgar or asked just for fun as long as they satisfied all other requirements.

We considered four sources of candidate questions which we hope to reflect the actual information need of the Internet users. First, thanks to the courtesy of Marcin Walas we were given access to user query logs of `Hipisek.pl`. The second source was ‘manual crawling’ around the QA communities. Similarly, we considered FAQ sites of several Inland Revenue offices. Lastly, we decided to abuse the auto-complete feature of Google and Bing search engines to gain insight into the questions that have actually been posed (it turns out that a number of users indeed ask natural language questions as search engine queries). The task was to enter word/words that typical questions start with and copy the suggestions. This could have led to some bias concerning the selection of the question-initial words. On the other hand, the mechanism seems to work surprisingly well and it is sufficient to give two words to obtain a lot of sensible questions. All of the questions that were decided as appropriate were subjected to orthographical and grammatical correction.

We considered manual translation of the TREC questions, as was done, e.g., in (Lombarović et al., 2011). We decided against this solution, since the TREC questions seem too much oriented on the American culture and geography for our purposes. Also the TREC datasets cotains mainly factoid questions while we wanted to create a balanced dataset containing both factoid and non-factoid questions.

2.2 Finding answers

We required from the answer documents to have included at least one passage (a couple of consecutive sentences) that contained the answer, i.e. that there was no necessity to construct an answer from information scattered across the document. This is because we assume the final version of the system will present such a passage to the user. We also required the answer-bearing passage to be comprehensive even when not paired with the question, e.g. if the question was about Linux, this name or its equivalent should appear in the passage rather than only general terms such as ‘*the system*’.

The set of candidate questions was given to linguists, which were asked to devote a couple of minutes per each question and try to find a satisfactory answer using search engines. They were asked to avoid typing the whole question as a query to prevent from favouring those documents

that contain questions.

For each candidate at most one answer document was found. Each answer document (a website) was downloaded as HTML files. We used the *Web As Corpus Toolkit* (Ziai and Ott, 2005) to clean up the files and remove boilerplate elements. The final output contained no mark-up and its structure was limited to plain text divided into paragraphs. Note that only those questions that the linguists were able to find an answer for have made their way to the final dataset.

2.3 Final collection

Ultimately, the set of questions paired with answers contains 598 entries. The statistics regarding source distribution is given as Table 1.

Source	Questions	Percentage
Hipisek.pl	236	39%
QA communities	98	16%
Search engines	244	41%
Revenue FAQ	20	3%
Overall	598	100%

Table 1: Distribution of sources in our dataset.

The collection contains the following types of questions (according to the expected answer type):

1. Factoid and list questions (Voorhees, 2004; Voorhees and Dang, 2005):
   - **person** — individual, group; *Who killed Osama bin Laden?*,
   - **location** — city, country, location-other; *In what city is the UAM located?*,
   - **organization** — band, company, institution, media, political-party; *What companies are listed within the WIG20?*,
   - **temporal** — date, period; *When was Albert Einstein born?*
   - **numerical** — count, money, numeric-other, size, temperature; *How many legs does a caterpillar have?*,
   - **other** — action, animal, artifact, band, color, disease, entity-other, expression, food, intellect-other, lake, language, plant, river, software, substance, vehicle, web-page; *Which dogs are aggressive?*, *What software can read epub files?*,
2. Non-factoid quesetions (Mizuno et al., 2007; Fukumoto, 2007):
   - **definition** — *What is X?*, *What does X mean?*, *Who is X?*,
   - **description** — *What powers does the president have?*,
   - **manner** — *how* questions; *How to start a business?*, *How to make a frappe coffe?*,
   - **reason** — *why* questions; *Why do cats purr?*, *How do I catch a cold?*.
3. Yes-no questions (Walas, 2012; Kanayama et al., 2012); *Is Lisbon in Europe?*.

Table 2 presents the number and the percentage of question types and subtypes in the the gathered collection.

Type	Count	Percent
Factoid and list questions	**267**	**44.65%**
– person	29	4.85%
– location	66	11.04%
– organization	15	2.51%
– temporal	37	6.19%
– numerical	45	7.53%
– other	75	12.54%
Non-factoid questions	**274**	**45.82%**
– definition	59	9.87%
– description	88	14.71%
– manner	68	11.36%
– reason	59	9.87%
Yes-no questions	**57**	**9.53%**

Table 2: Types of questions.

To perform a reliable evaluation of the system, we had to index a lot more data than just the answers to our 598 test questions. We acquired also several collections to serve as 'distractors' and a source of possible answers, namely:

- Polish Wikipedia (using dump from 22 January 2013) — 956 000 documents.
- A collection of press articles from *Rzeczpospolita* (Weiss, 2008) — 180 000 documents.
- Three smaller corpora: KPWr (Broda et al., 2012), CSEN and CSER (Marcińczuk and Piasecki, 2011) — 3 000 documents.

3 Evaluation metrics

The evaluation was based on the following metrics:

- **answers at n-th cutoff** (a@n) (Monz, 2003) — *relevant documents recall*; a fraction of questions for which the relevant document was present in the first n documents returned by the search engine;
- **mean reciprocal rank** (MRR) — an average of the query reciprocal ranks[1] MRR is used to compare re-ranking algorithms. The higher MRR is, the higher in the ranking the relevant documents are.

4 Baseline information retrieval

As a basis for search engine we selected an open source search platform called *Solr* (The Apache Software Foundation, 2013a). *Solr* indexes large collection of documents and provides: full-text search, rich query syntax, document ranking, custom document fields and terms weighting. It was also shown that Lucene (the retrieval system underlying Solr) performs no worse for QA than other modern Information Retrieval systems (Tellex et al., 2003).

In the baseline approach we used an existing tool called *Web as Corpus ToolKit* (Adam Kilgarriff and Ramon Ziai and Niels Ott, 2013) to extracted plain text from the collection of HTML documents. Then, the text was tagged using WCRFT tagger (Radziszewski, 2013) and their base forms were indexed in the Solr.

To fetch a ranked list of documents for a query we used a default search ranking algorithm implemented in the *Lucene* that is a combination of Boolean Model (BM) with refined Vector Space Model (VSM). BM is used to fetch all documents matching the boolean query. Then, VSM is applied to rank the answer documents. The detailed formula used to compute the ranking score is presented in (The Apache Software Foundation, 2013b). The formula includes following factors:

- fraction of query terms present in the document — documents containing more query terms are scored higher than those with fewer,
- query normalizing factor — to make the score comparable between queries,
- document term frequency — documents containing more occurrences of query terms receive higher scores,
- inverse document frequency — common terms (present in many documents) have lower impact on the score,
- term boosting factor — weight specified in the query can be used to increase importance of selected terms (not used in our approach),
- field boosting factor — some fields might be more important than others (not used by us),
- field length normalization factor — shorter fields obtain higher scores.

Figure 1 presents all steps of *question analysis*. First, a question is tagged with WCRFT tagger. All punctuation marks and words from a stoplist (including 145 prepositions) are discarded. We assumed that in most cases the answer have a form of a statement and does not mimic question structure. The remaining words are used in a query, formed as a boolean disjunction of the base forms.

The a@n and MRR values for the baseline configuration are presented in Table 3. We measured the a@n for several distinct values of n between 1 and 200 (this is an estimated maximum number of documents which can be effectively processed during re-ranking). The a@n ranges from 26% for $n = 1$ to 87% for $n = 200$. This means than only for 26% questions the relevant document was on the first position in the ranking. In the reported tests all non-stop words from the question were used to form a query. We tested also several modification of the heuristic for query term selection proposed in (Paşca, 2003), but the results were lower.

5 Proximity-based re-ranking

Lucene default ranking algorithm does not take into consideration proximity of query terms in the documents. This leads to favouring longer documents as they are more likely to contain more query terms. However such documents can describe several different topics not related to the question. Ranking of longer documents cannot be decreased by default, as they might contain an answer. A possible solution is to analyse query term

[1] A reciprocal rank for a query is equal to $\frac{1}{K}$, where K is the position of first relevant document in the ranking.

1.	**Input:**	*Co można odliczyć od podatku?* (“What can be deducted from tax?”)
2.	**Tagging:**	*co można odliczyć od podatek ?* (base forms)
3.	**Filtering:**	*można odliczyć od podatek* (“can”, “deduct”, “tax”)
4.	**Query:**	`base:można OR base:odliczyć OR case:od OR base:podatek`

Figure 1: Steps of processing for a sample question.

	a@n
n	**baseline**
1	26.09%
5	52.17%
10	62.04%
20	70.57%
50	76.76%
100	82.61%
200	87.29%
MRR	0.3860

Table 3: a@n and MRR for baseline configuration of information retrieval.

proximity inside the documents. We have evaluated two approaches to utilising term proximity in re-ranking.

5.1 Maximum Cosine Similarity Weighting

Maximum Cosine Similarity Weighting (MCSW) is based on the idea of using the same ranking scheme as in the retrieval component, but applied to short passages, not whole documents. Every document is divided into continuous blocks of k sentences. For every block we compute the cosine similarity between a vector representing the block and a vector representing a query. Standard *tf-idf weighting* (Manning et al., 2008) and cosine measure are used. A document is assigned the maximum per-block cosine similarity that was encountered. Several block sizes (k from 1 to 5) were tested producing very similar results, thus we report results only for $k = 1$. The final document score is computed as follows:

$$score'(d) = \frac{score(d)}{\arg\max_{d \in D} score(d)} \cdot \frac{mcs(d)}{\arg\max_{d \in D} mcs(d)} \quad (1)$$

where:

- D, ordered list od documents returned from search engine for a query,
- $score(d)$, score for document d returned by *Solr*,
- $mcs(d)$, maximum cosine similarity for document d.

5.2 Minimal Span Weighting

Monz (2003) presented a simple method for weighting based on a minimal text span containing all terms from a query that occur in the document. The re-ranking score combines the original score with MSW score and is computed as follows:

$$score''(d) = score(d) * \lambda + (1 - \lambda) \cdot \left(\frac{|q \cap d|}{|s|}\right)^{\alpha} \cdot \left(\frac{|q \cap d|}{|q|}\right)^{\beta} \quad (2)$$

where:

- q, set of query terms,
- s, the shortest text fragment containing all query terms occurring in the document,
- λ, α, β, significance weights for the respective factors (we used default values (Monz, 2003), i.e. $\lambda = 0.4$, $\alpha = 0.125$, $\beta = 1$)

5.3 Evaluation

The a@n and MRR values for MCSW and MSW are presented in Table 4. For both methods we noticed a small improvement. The increase of MRR values for both methods indicates that the average position of the relevant documents in the ranking was improved. The a@n was improved by up to 12 percentage points for MCSW and $n = 1$. The lower improvement for MSW might be caused by the assumption that the minimal span must contain all query terms occurring in the document.

This may result in very long spans covering almost complete documents. In the case of MCSW we force the sentence-based segmentation and mostly only fractions of the covered query terms influence MCSW score.

	a@n		
n	**baseline**	**MCSW** ($k = 1$)	**MSW**
1	26.09%	38.63%	33.95%
5	52.17%	63.04%	58.36%
10	62.04%	71.24%	68.73%
20	70.57%	78.43%	74.58%
50	76.76%	83.95%	79.93%
100	82.61%	86.45%	84.11%
200	87.29%	87.29%	87.29%
MRR	0.3860	0.5007	0.4555

Table 4: a@n and MRR for baseline information retrieval with reranking.

In addition, the proximity-based ranking algorithms can be used to extract the most relevant document fragments as answers instead of presenting the whole document. According to (Lin et al., 2003), users prefer paragraph-level chunks of text with appropriate answer highlighting.

Despite the observed improvements, the results are still below our expectations. If we assume that user reads up to 10 answers for a question (a typical number of results displayed on a single page in many web search engines), the top a@n will be about 70%. This means that we will not provide any relevant answer for 3 out of 10 questions. According to (Monz, 2003), results for English reported for TREC sets are between 73% and 86% for a@10. Thus, further improvement in reranking is necessary.

6 Conclusion

We presented a preliminary results for a baseline information retrieval system and the simple proximity-based re-ranking methods in the context of a Open Domain Question Answering task for Polish. The evaluation was performed on a corpus of 598 questions and answers, collected from a wide range of questions asked by Internet users (i.e. search engines, `Hipisek.pl`, QA communities and Revenue FAQ). The collection covers major types of questions including: factoid, list, non-factoid and yes-no questions.

The a@n of the baseline IR system (*Solr*) configuration ranges from 26% for $n = 1$ to 87% for $n = 200$ top documents considered. Our queries consisted of base forms of all question words except words from a stoplist. Several heuristics for query term selection inspired by the one proposed in (Monz, 2003) produced lower results. This can be explained by the properties of the ranking algorithm used in *Solr* — the number of terms covered and their total frequency in a document are important factors. For $n = 10$ (a typical single page in a Web search) we obtained 62% a@n. Two re-ranking methods based on query term proximity were applied. For both methods we obtained a noticeable improvement up to 12 percentage points of a@n for $n = 1$ and 9 percentage points for $n = 10$. Nevertheless, the results are still slightly lower than in the case of systems built for English, e.g., (Monz, 2003). However, results reported by Monz were obtained on the TREC datasets, which contain mostly factoid and list questions. Our datasets includes also non-factoid and yes-no questions which are more difficult to deal with. The comparison with *Hipisek* is difficult as no results concerning ranking precision were not reported. Moreover, *Hipisek* was focused on selected subclasses of questions.

We plan to extend the information retrieval model on the level of document fetching and re-ranking. We want to utilize plWordNet 2.0 (the Polish wordnet)[2] (Maziarz et al., 2012), tools for proper names (Marcińczuk et al., 2013) and semantic relations recognition (Marcińczuk and Ptak, 2012), dependency[3] and shallow syntactic parsers. More advanced but also more time-consuming tools will be used to select relevant passages in the documents fetched by the presented information retrieval module.

Acknowledgements The work was funded by the European Union Innovative Economy Programme project NEKST, No POIG.01.01.02-14-013/09.

We would like to thank Marcin Walas for sharing the collection of historic questions that have been posed to the Hipisek system.

[2] `http://www.nlp.pwr.wroc.pl/plwordnet/`

[3] `http://zil.ipipan.waw.pl/PolishDependencyParser`

References

Adam Kilgarriff and Ramon Ziai and Niels Ott. 2013. Web as Corpus ToolKit. March.

Bartosz Broda, Michał Marcińczuk, Marek Maziarz, Adam Radziszewski, and Adam Wardyński. 2012. KPWr: Towards a Free Corpus of Polish. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of LREC'12*, Istanbul, Turkey. ELRA.

Jennifer Chu-Carroll, James Fan, Branimir Boguraev, David Carmel, Dafna Sheinwald, and Chris Welty. 2012. Finding needles in the haystack: Search and candidate generation. *IBM Journal of Research and Development*, 56(3):6.

David A. Ferrucci. 2012. Introduction to "This is Watson". *IBM Journal of Research and Development*, 56(3):1.

Junichi Fukumoto. 2007. Question Answering System for Non-factoid Type Questions and Automatic Evaluation based on BE Method. In Kando and Evans (Kando and Evans, 2007), pages 441–447.

David Gondek, Adam Lally, Aditya Kalyanpur, J. William Murdock, Pablo Ariel Duboué, Lei Zhang, Yue Pan, Zhaoming Qiu, and Chris Welty. 2012. A framework for merging and ranking of answers in DeepQA. *IBM Journal of Research and Development*, 56(3):14.

Hiroshi Kanayama, Yusuke Miyao, and John Prager. 2012. Answering Yes/No Questions via Question Inversion. In *COLING*, pages 1377–1392.

Noriko Kando and David Kirk Evans, editors. 2007. *Proceedings of the Sixth NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering, and Cross-Lingual Information Access*, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan, May. National Institute of Informatics.

Adam Lally, John M. Prager, Michael C. McCord, Branimir Boguraev, Siddharth Patwardhan, James Fan, Paul Fodor, and Jennifer Chu-Carroll. 2012. Question analysis: How Watson reads a clue. *IBM Journal of Research and Development*, 56(3):2.

Jimmy J. Lin, Dennis Quan, Vineet Sinha, Karun Bakshi, David Huynh, Boris Katz, and David R. Karger. 2003. What Makes a Good Answer? The Role of Context in Question Answering. In *Human-Computer Interaction INTERACT '03: IFIP TC13 International Conference on Human-Computer Interaction, 1st–5th September 2003, Zurich, Switzerland*. IOS Press.

Tomislav Lombarović, Jan Šnajder, and Bojana Dalbelo Bašić. 2011. Question classification for a Croatian QA system. In *Proceedings of the 14th international conference on Text, speech and dialogue*, TSD'11, pages 403–410, Berlin, Heidelberg. Springer-Verlag.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

Michał Marcińczuk and Maciej Piasecki. 2011. Statistical Proper Name Recognition in Polish Economic Texts. *Control and Cybernetics*, 40(2).

Michał Marcińczuk and Marcin Ptak. 2012. Preliminary Study on Automatic Induction of Rules for Recognition of Semantic Relations between Proper Names in Polish Texts. In Petr Sojka, Aleš Horák, Ivan Kopecek, and Karel Pala, editors, *Text, Speech and Dialogue*, volume 7499 of *Lecture Notes in Computer Science*, pages 264–271. Springer Berlin / Heidelberg. 10.1007/978-3-642-32790-2_32.

Michał Marcińczuk, Jan Kocoń, and Maciej Janicki. 2013. Liner2 – A Customizable Framework for Proper Names Recognition for Polish. In Robert Bembenik, Lukasz Skonieczny, Henryk Rybinski, Marzena Kryszkiewicz, and Marek Niezgodka, editors, *Intelligent Tools for Building a Scientific Information Platform*, volume 467 of *Studies in Computational Intelligence*, pages 231–253. Springer Berlin Heidelberg.

Marek Maziarz, Maciej Piasecki, and Stanisław Szpakowicz. 2012. Approaching plWordNet 2.0, January.

Junta Mizuno, Tomoyosi Akiba, Atsushi Fujii, and Katunobu Itou. 2007. Non-factoid Question Answering Experiments at NTCIR-6: Towards Answer Type Detection for Realworld Questions. In Kando and Evans (Kando and Evans, 2007), pages 487–492.

Christof Monz. 2003. *From Document Retrieval to Question Answering*. Phd thesis, Universiteit van Amsterdam.

Marius Paşca. 2003. *Open-Domain Question Answering from Large Text Collections*. University of Chicago Press.

Adam Radziszewski. 2013. A Tiered CRF Tagger for Polish. In Robert Bembenik, Łukasz Skonieczny, Henryk Rybiński, Marzena Kryszkiewicz, and Marek Niezgódka, editors, *Intelligent Tools for Building a Scientific Information Platform*, volume 467 of *Studies in Computational Intelligence*, pages 215–230. Springer Berlin Heidelberg.

Stefanie Tellex, Boris Katz, Jimmy Lin, Aaron Fernandes, and Gregory Marton. 2003. Quantitative evaluation of passage retrieval algorithms for question answering. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, SIGIR '03, pages 41–47, New York, USA. ACM.

The Apache Software Foundation. 2013a. Apache Solr. March.

The Apache Software Foundation. 2013b. Implementation of Lucene default search and ranking algorithm. March.

Ellen M. Voorhees and Hoa Trang Dang. 2005. Overview of the TREC 2005 Question Answering Track. In Ellen M. Voorhees and Lori P. Buckland, editors, *TREC*, volume Special Publication 500-266. National Institute of Standards and Technology (NIST).

Ellen M. Voorhees. 2001. The TREC question answering track. volume 7, pages 361–378, New York, USA, December. Cambridge University Press.

E. Voorhees. 2004. Overview of the TREC 2004 Question Answering Track. In Ellen M. Voorhees and Lori P. Buckland, editors, *Proceedings of the 13th Text Retrieval Conference (TREC)*, volume Special Publication 500-261, pages 52–62. National Institute of Standards and Technology (NIST).

Marcin Walas and Krzysztof Jassem. 2011. Named Entity Recognition in a Polish Question Answering System. In *Proceedings of Intelligent Information Systems*, pages 181–191.

Marcin Walas. 2012. How to Answer Yes/No Spatial Questions Using Qualitative Reasoning? In Alexander F. Gelbukh, editor, *CICLing (2)*, volume 7182 of *Lecture Notes in Computer Science*, pages 330–341. Springer.

Dawid Weiss. 2008. Korpus Rzeczpospolitej. [on-line] `http://www.cs.put.poznan.pl/dweiss/rzeczpospolita`. Corpus of text from the online edtion of Rzeczypospolita.

Ramon Ziai and Niels Ott, 2005. *Web as Corpus Toolkit: User's and Hacker's Manual*. Lexical Computing Ltd., Brighton, UK. Manual for version pre3.