Sentence and Word Embedding Employed in Open Question-Answering

Marek Medveď and Aleš Horák
Natural Language Processing Centre, Faculty of Informatics, Masaryk University,
Botanická 68a, 602 00, Brno, Czech Republic

Keywords: Question Answering, Word Embedding, Word2vec, AQA, Simple Question Answering Database, SQAD.

Abstract: The Automatic Question Answering, or AQA, system is a representative of open domain QA systems, where the answer selection process leans on syntactic and semantic similarities between the question and the answering text snippets. Such approach is specifically oriented to languages with fine grained syntactic and morphologic features that help to guide the correct QA match. In this paper, we present the latest results of the AQA system with new word embedding criteria implementation. All AQA processing steps (question processing, answer selection and answer extraction) are syntax-based with advanced scoring obtained by a combination of several similarity criteria (TF-IDF, tree distance, ...). Adding the word embedding parameters helped to resolve the QA match in cases, where the answer is expressed by semantically near equivalents. We describe the design and implementation of the whole QA process and provide a new evaluation of the AQA system with the word embedding criteria measured with an expanded version of Simple Question-Answering Database, or SQAD, with more than 3,000 question-answer pairs extracted from the Czech Wikipedia.

1 INTRODUCTION

Open domain question answering (QA) systems do not pose any limit on the content of the question, they rather aim to serve as “next generation” search systems (Etzioni, 2011). The capabilities of QA systems are, of course, limited by their actual information source – either a (form of a) *knowledge base* or a (set of) text document(s)/corpora. A knowledge base is formed by structured data that contain additional information which is added by either manual or automatic annotation. Known representatives of this group are Knowledge Graph (Singhal, 2012), DBpedia (Lehmann et al., 2015) and WikiQA (Yang et al., 2015). A QA text corpus comprises generally plain text data (possibly with linguistic annotations of words) separated to *questions* and *answers* (see e.g. SQuAD (Rajpurkar et al., 2016)). Such text corpus usually first needs to be processed by text mining techniques to obtain all available information from the text to be able to build an underlying (structured) data source containing all candidate answers.

Most of the recent QA systems in both categories (Fader et al., 2014; Sun et al., 2015; Yih et al., 2014; Xiong et al., 2017; Wang et al., 2017) are based on data resources related to the mainstream languages, usually English. They offer the best coverage of questioned topics – either in the form of large curated knowledge bases or huge corpora (Pomikálek et al., 2012) offering invaluable statistics on multiword contexts and computational semantics models. However, the dependence on such resources means that the system capabilities are not fully transferable to less-resourced languages without a substantial decrease in the final system accuracy.

In the following text, we present the details of a new version of question answering system named AQA (Automatic Question Answering) (Medveď and Horák, 2016), which was designed for languages that have the advantage (in this case) of rich flectional and morphological structures providing extra guidance for question-answer selection. AQA introduces new techniques that aim to improve QA over these less-resourced languages. The system is developed currently for the Czech language, but the employed syntax-based techniques apply to many other morphologically rich languages, e.g. most languages from the Slavonic language family.

In this paper, we describe the design and implementation of the whole QA process of the AQA system and provide an evaluation of applying new criteria for the answer selection and extraction based on word embeddings. We also present a new version of QA evaluation database SQAD (Simple Question-Answering Database (Horák and Medveď, 2014))

Medveď, M. and Horák, A.
Sentence and Word Embedding Employed in Open Question-Answering.
DOI: 10.5220/0006595904860492
In *Proceedings of the 10th International Conference on Agents and Artificial Intelligence (ICAART 2018) - Volume 2*, pages 486-492
ISBN: 978-989-758-275-2

```
a)    word/token          lemma       tag
      <s>
      Jak                 jak         k6eAd1
      se                  sebe        k3xPyFc4
      jmenuje             jmenovat    k5eAaImIp3nS
      světově             světově     k6eAd1
      nejrozšířenější     rozšířený   k2eAgFnSc1d3
      hra                 hra         k1gFnSc1
      na                  na          k7c4
      hrdiny              hrdina      k1gMnPc4
      <g/>
      ?                   ?           kIx.
      </s>

b)    Dungeons & Dragons
c)    Nejrozšířenější světově hranou RPG hrou na hrdiny pak je Dungeons & Dragons.
d)    https://cs.wikipedia.org/wiki/Hra_na_hrdiny
```

Figure 1: SQAD Q/A example: a) the analyzed question of "*Jak se jmenuje světově nejrozšířenější hra na hrdiny?* " (What is the name of the world's most spread role-playing game?), b) the answer, c) the answer sentence of "*Nejrozšířenější světově hranou RPG hrou na hrdiny pak je Dungeons & Dragons.* (The world's most widely played RPG role-playing game is then Dungeons & Dragons.), and d) the Wikipedia document containing the answer.

with more than 3,000 question-answer pairs extracted from the Czech Wikipedia. In this version, the underlying texts were expanded to full texts of the respective Wikipedia articles reaching 6 million tokens. The previous version contained reduced answer contexts with about 280,000 tokens. The expanded SQAD dataset is then used for evaluation of the new AQA features.

2 THE AQA QUESTION-ANSWERING SYSTEM

In the following text we describe the AQA system modules that are employed in extracting a concrete answer to a given question. We mainly focus on the Answer selection module where the word and sentence embedding features are exploited.

2.1 Question Processor

The first phase in the QA processing by the AQA system is denoted as the Question processor (see Figure 2). First, the input texts (both the question and candidate answers) are processed by the Czech morphological analyser Majka (Šmerk, 2009; Jakubíček et al., 2011) and disambiguated by the DESAMB tagger (Šmerk, 2010). An example of a question enriched by lexical and morphological information is presented in Figure 1.

In further processing, several question features are extracted from the text – the question syntactic tree (using the SET parser (Kovář et al., 2011)), the question type, the question main verb/subject, named entities, birth/death dates and birth/death places[1] (following the structure of the question syntactic tree, see Figure 3).

The SET parsing process is based on pattern matching rules for link extraction with probabilistic tree construction. For the purposes of the AQA system, the original SET grammar was supplemented by special rules for question type recognition.

The question processor of the AQA system recognizes these question types:[2]

- **Functional word questions** are all non Wh* questions that usually start with verb.
- **When questions** focus on an exact time or a time span.
- **Where questions** focus on a place.
- **Which questions** where the focus lies on the noun phrase following the "Which" word.
- **Who questions** ask about some entity.
- **Why questions** want to find out the reason or explanation.
- **How questions** ask for explanation or number ("how much/many").

[1] *Birth/death date* and *birth/death place* are features of specific sentences where the birth/death date and birth/death place are present in the text just after a personal name, usually in parentheses.

[2] See e.g. (Li and Roth, 2002) for detailed discussion on question classification.

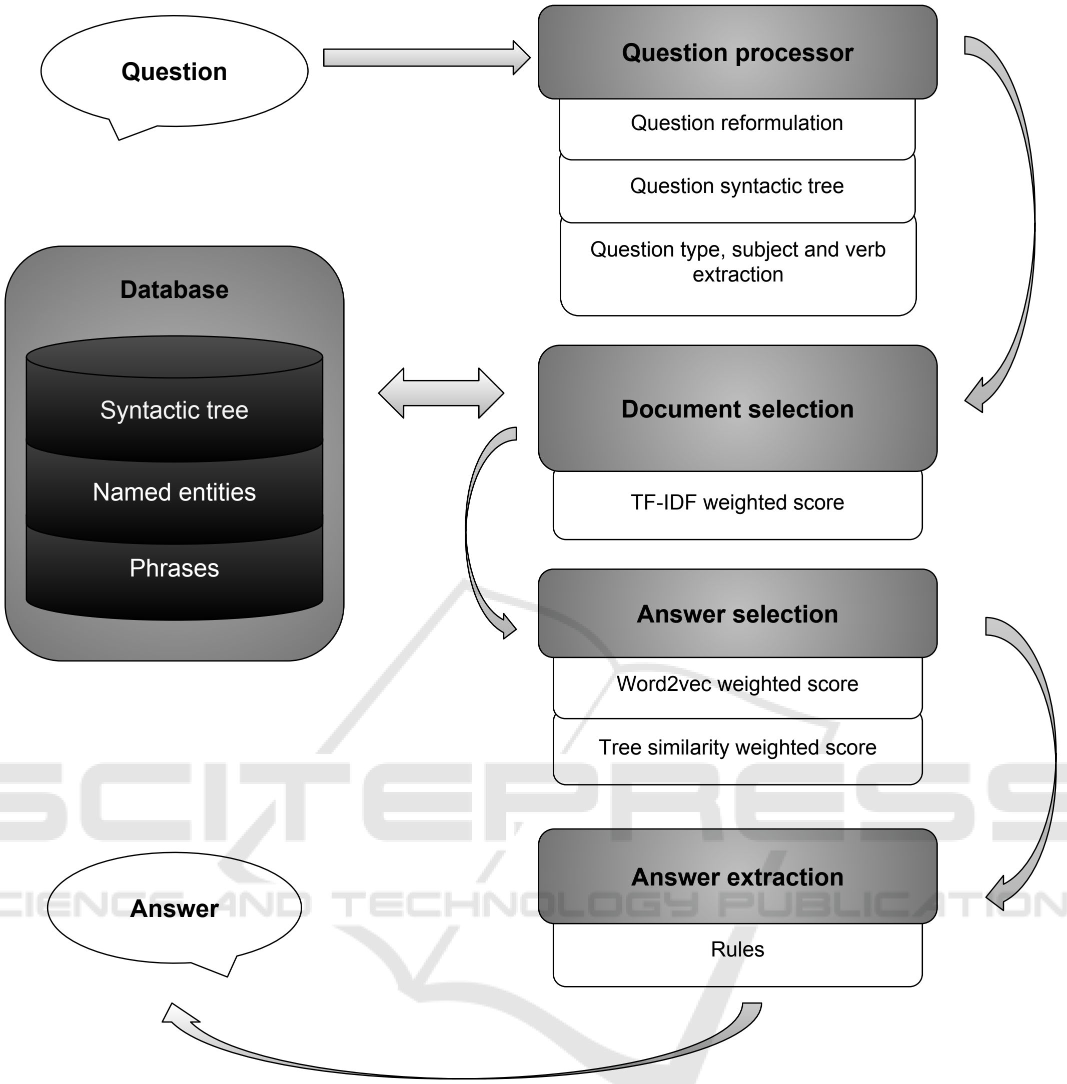


Figure 2: The AQA system modules.

- **What questions** for general questions.

The answer scoring methods employ named entity recognition (NER) to be able to match the respective questioned entity types with factual data in the underlying texts. The AQA system recognizes three named entity types: a *place*, an *agent* and an *art work*. In the current version, AQA uses the Stanford Named Entity Recognizer (Finkel et al., 2005) with a model trained on the Czech Named Entity Corpus 1.1 (Ševčíková et al., 2014) and the Czech DBpedia data.

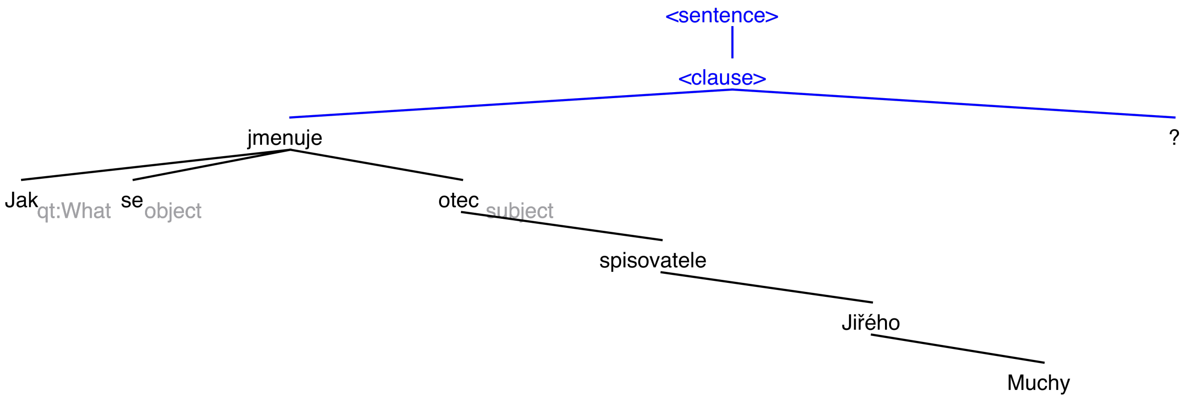
These features are essential in the answer selection and extraction parts.

2.2 Answer Selection

In this section, we describe the exploitation of the word and sentence embeddings approach used to extract a correct sentence(s) from the knowledge base. The decision process uses a list of sentences ordered by the resulting confidence score denoting the probability of a successful match between the question and the sentence answer. The score is computed as a weighted sum of attribute similarities between the question and the answer.

The extraction of a possible answer sentence is divided into two steps:

- *Document Selection:* after the system has obtained all the possible pieces of information about



question type:	*What*
subject:	*otec*
main verb:	*jmenuje*
phrases:	*Jiřího Mouchy* (noun phrase, genitive, singular, masculine, capitalized)
	spisovatele Jiřího Mouchy (noun phrase, genitive, singular, masculine)
	otec spisovatele Jiřího Mouchy (noun phrase, nominative, singular, masculine)

Figure 3: Syntactic tree of the sentence “Jak se jmenuje otec spisovatele Jiřího Mouchy?”(What is the name of the father of Jiří Moucha, the writer?) with the question type, the question main verb/subject, named entities and phrases.

the question, it can focus on the selection of the best answering document among all the documents in the database. For this process, we employ the TF-IDF implementation from the **gensim** library (Řehůřek and Sojka, 2010). The system creates a similarity matrix among all documents in the database (document term matrix). The most promising document is then selected by cosine distance between the question and the best document from the similarity matrix.

- *Answer Sentence Selection:* for this subtask, we have implemented two sentence similarity scoring computation methods based on the gensim modules of the word embedding *Word2Vec* technique (Mikolov et al., 2013) and the phrase embedding *Doc2Vec* technique (Le and Mikolov, 2014). The sentence similarity scoring distances then help to order all possible answer sentences.

  Doc2Vec (document/sentence embeddings) modifies the word2vec algorithm to unsupervised learning of continuous representations for larger blocks of text, such as sentences, paragraphs or entire documents.

  The *Doc2Vec* module was used to train a model of sentence vector space where each sentence is represented by one vector. Before the training, the sentences are lemmatized (each word is substituted by its base form) and stop-words[3] are removed.

  The final *Doc2Vec* module compares the embedding vector of the question (also lemmatized and filtered by the stop-word list) to all candidate sentence embedding vectors to identify the best answer sentence. However, according to the evaluation results, this method was not the most accurate.

  The second method is based on a combination of individual word embeddings (by *Word2Vec*) based on syntactic structure of the respective sentence. The vector space model in this case is trained with all words in the documents except stop-words. These sentence scores are then computed by the following steps:

  a) for each phrase in the sentence and the question create a phrase vector by sum of its word-vector representations (example in Figure 4),
  b) for each question phrase calculate the cosine similarity with each answer phrase and find the maximal cosine similarity value (illustration in Figure 5),
  c) the average of the maximal cosine similarities between question and answer phrases forms the final sentence score.

  In the final step, the sentence similarity score is combined with tree distance score which computes tree distance mappings between each words pair in question-answer noun phrases. The system

[3] The stop-word list was extracted from a large corpus of Czech, `cztenten` (Jakubíček et al., 2013), and is used to remove the most frequent words, which are not important for semantic representation of a sentence.

sentence	*Jak se jmenuje otec spisovatele Jiřího Mouchy?*
word	*word vector*
otec	0 0 0 1 0 0 0
spisovatele	0 0 0 0 1 0 0
Jiřího	0 0 0 0 0 1 0
Mouchy	0 0 0 0 0 0 1
phrase	*phrase vector*
Jiřího Mouchy	0 0 0 0 0 1 1
spisovatele Jiřího Mouchy	0 0 0 0 1 1 1
otec spisovatele Jiřího Mouchy	0 0 0 1 1 1 1

Figure 4: Phrase vectors of the sentence “Jak se jmenuje otec spisovatele Jiřího Mouchy?”(What is Jiří Moucha father’s name?).

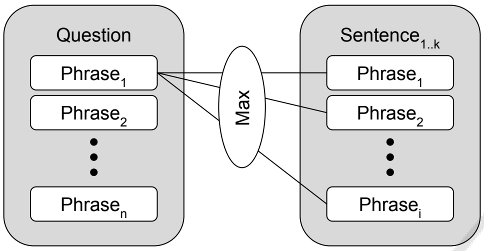


Figure 5: Maximal cosine similarity value between the question and a candidate sentence.

resulting score identifies the best scored sentence as the most probable sentence containing the answer.

To the best of our knowledge this approach of transformation of word vectors and syntactic structure into a phrase vector representation has never been introduced in previous work.

According to the evaluation (see Section 3), this method outperforms the *Doc2Vec* approach.

2.3 The Answer Extraction Module

The final step of the AQA processing is accomplished by the Answer extraction module where a particular part of each sentence is extracted and declared as the final answer. This part of the system works on the best scored candidate answer sentences that were picked up by the Answer selection module.

The final answer for the asked question is extracted according to the following three factors:

- *Answer Named Entities:* within the knowledge base creation process, AQA extracts the supported named entities of three types, i.e. Place, Agent and ArtWork. The Answer extraction module then maps the question focus to the extracted answer named entities. In this process, AQA also excludes named entities that are present in the question to avoid an incorrect answer. This is the first attempt to get a concrete answer.
- *Answer Numeric Quantity:* as a special case of named entity, numeric quantities are identified in the answer texts and serve as the extracted answer for question classes that require (or allow) numeric result.
- *Answer Noun Phrases:* in case the previous step fails to find an answer, the system selects one phrase from the phrase list as the answer according to the *Question focus*.[4]

3 EVALUATION

Within the evaluation process, we have used both the SQAD v1.0 database (Horák and Medveď, 2014) and its new **expanded** version, denoted as **SQAD v1.1**. Both SQAD versions contain the same number of 3,301 entries of a question, the answer to this question and the answer text. The previous version used a rather small set of answer sentences as the knowledge database for the answer selection process, the answer sentences were chosen as the minimal context from which the answer can be derived. The current expanded version v1.1 uses the whole set of Wikipedia documents used in the SQAD preparation phase rather than just the closest-context paragraphs that have been used in SQAD v1.0. The SQAD knowledge database, which is searched in the answer selection process, has thus been substantially enlarged – see the statistics of sentence and token numbers in Table 3.

In the evaluation, the AQA knowledge database was built from all the answer texts from SQAD, and all 3,301 questions were answered by the AQA system. There are three possible types of a match between the AQA answer and the expected answer:

[4] At the beginning, each question is classified by a question type according to the sentence structure and the words present in the question. The question type then determines what is the actual focus of the question in terms of expected entity type.

Table 1: Evaluation of the AQA system on the expanded SQAD v1.1 database.

a) syntax oriented Word2Vec sentence representation

	Answer extraction	in %
Match	1,257	38.08 %
Partial match	270	8.18 %
Mismatch	1,774	53.74 %
Total	3,301	100.00 %

b) gensim Doc2Vec trained on sentences

	Answer extraction	in %
Match	875	26.51 %
Partial match	184	5.57 %
Mismatch	2,242	67.92 %
Total	3,301	100.00 %

Table 2: Evaluation of the AQA system on the SQAD v1.0 database (syntax oriented Word2Vec sentence representation).

	Answer extraction	in %
Match	1,645	49.83 %
Partial match	322	9.75 %
Mismatch	1,334	40.41 %
Total	3,301	100.00 %

- a (full) *Match* – the first provided answer is equal to the expected answer;
- a *Partial match* – the provided answer is not an exact phrase match, some words are either missing or redundant;
- a *Mismatch* – incorrect or no answer produced.

For the results on the original SQAD v1.0 database see Table 2, for the results on expanded SQAD v1.1 database see Table 1a). The SQAD v1.0 results are presented for a comparison with the best evaluation published in (Medveď and Horák, 2016). The number of correct answers using the current method has increased by **9.63 %**. In the evaluation of the expanded SQAD 1.1, the system has to identify the answer sentence in a much larger number of sentences (60× more), which is the main reason for a lower proportion of correct answers (38 %) in this case.

The presented best results are based on the second method using sentence scores obtained by syntax motivated combinations of word embeddings (see Section 2.2). As a comparison, the Table 1b) shows the same evaluation of the system using the *Doc2Vec* method. We can see that in same settings our syntax motivated approach outperforms the *Doc2Vec* approach by **11.57 %** in the full match category.

Table 3: Numbers of sentences and tokens (words and punctuation) in SQAD v1.0 and the expanded version SQAD v1.1.

database	SQAD v1.0	exp. SQAD v1.1
Sentences	4,442	279,921
Tokens	97,461	6,185,697

4 CONCLUSIONS AND FUTURE DIRECTIONS

In this paper, we have presented the latest development of the AQA syntax-based question answering system which introduces new techniques for less-resourced languages based on rich flectional and morphological structures.

We have performed an evaluation of two approaches to exploitation of sentence vector representations, or sentence embeddings, within the answer selection part of the QA process.

The results show that our syntax motivated approach using Word2Vec phrasal combinations outperforms the general Doc2Vec model by 11.6 % and show that the embeddings based on syntax knowledge are more adequate for syntactically rich languages.

We have also presented a new expanded version of the Simple Question-Answering Database (SQAD v1.1), which now includes 6 million tokens of underlying texts formed by full Wikipedia documents related to the set of questions. The current version of AQA is evaluated both with the SQAD v1.0 and the expanded SQAD v1.1 for comparison showing an expected but not radical decrease from 49 % to 38 % while searching through 60× larger knowledge base.

For future work, we plan to incorporate more techniques for flectional and morphologically rich languages into AQA to improve question-answer selection.

ACKNOWLEDGEMENTS

This work has been partly supported by the Grant Agency of CR within the project 15-13277S.

REFERENCES

Etzioni, O. (2011). Search needs a shake-up. *Nature*, 476(7358):25–26.

Fader, A., Zettlemoyer, L., and Etzioni, O. (2014). Open question answering over curated and extracted knowledge bases. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1156–1165. ACM.

Finkel, J. R., Grenager, T., and Manning, C. (2005). Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 363–370, Stroudsburg, PA, USA. Association for Computational Linguistics.

Horák, A. and Medved', M. (2014). SQAD: Simple question answering database. In *Eighth Workshop on Recent Advances in Slavonic Natural Language Processing*, pages 121–128, Brno. Tribun EU.

Jakubíček, M., Kilgarriff, A., Kovář, V., Rychlý, P., and Suchomel, V. (2013). The tenten corpus family. In *7th International Corpus Linguistics Conference*, pages 125–127.

Jakubíček, M., Kovář, V., and Šmerk, P. (2011). Czech Morphological Tagset Revisited. *Proceedings of Recent Advances in Slavonic Natural Language Processing 2011*, pages 29–42.

Kovář, V., Horák, A., and Jakubíček, M. (2011). Syntactic Analysis Using Finite Patterns: A New Parsing System for Czech. In *Human Language Technology. Challenges for Computer Science and Linguistics*, pages 161–171, Berlin/Heidelberg.

Le, Q. V. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *ICML*, volume 14, pages 1188–1196.

Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., Hellmann, S., Morsey, M., Van Kleef, P., Auer, S., et al. (2015). DBpedia – a large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web*, 6(2):167–195.

Li, X. and Roth, D. (2002). Learning question classifiers. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7. Association for Computational Linguistics.

Medved', M. and Horák, A. (2016). AQA: Automatic Question Answering System for Czech. In *International Conference on Text, Speech, and Dialogue*, pages 270–278. Springer.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. In *Proceedings of Workshop at the International Conference on Learning Repre- sentation*, Scottsdale, USA.

Pomikálek, J., Jakubíček, M., and Rychlý, P. (2012). Building a 70 billion word corpus of English from ClueWeb. In *LREC*, pages 502–506.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). Squad: 100, 000+ questions for machine comprehension of text. *CoRR*, abs/1606.05250.

Řehůřek, R. and Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA.

Ševčíková, M., Žabokrtský, Z., Straková, J., and Straka, M. (2014). Czech named entity corpus 1.1.

Singhal, A. (2012). Introducing the knowledge graph: things, not strings.

Šmerk, P. (2009). Fast Morphological Analysis of Czech. In *Proceedings of 3rd Workshop on Recent Advances in Slavonic Natural Language Processing, RASLAN 2009*, Brno.

Šmerk, P. (2010). *Towards Computational Morphological Analysis of Czech*. PhD thesis, Masaryk University, Brno, Czech Republic.

Sun, H., Ma, H., Yih, W.-t., Tsai, C.-T., Liu, J., and Chang, M.-W. (2015). Open domain question answering via semantic enrichment. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1045–1055. ACM.

Wang, W., Yang, N., Wei, F., Chang, B., and Zhou, M. (2017). Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 189–198.

Xiong, C., Zhong, V., and Socher, R. (2017). Dynamic coattention networks for question answering. *International Conference on Learning Representations*, abs/1611.01604.

Yang, Y., Yih, W.-t., and Meek, C. (2015). WikiQA: A challenge dataset for open-domain question answering. In *Proceedings of the Conference on Empirical Methods for Natural Language Processing, EMNLP 2015*, pages pp.2013–2018.

Yih, W.-t., He, X., and Meek, C. (2014). Semantic parsing for single-relation question answering. In *ACL (2)*, pages 643–648. The Association for Computational Linguistics.