Answer Extraction for Question Answering Game Application

Desmond Darma Putra, Volha Petukhova and Dietrich Klakow

Saarland University, Spoken Language Systems, Saarbrücken, Germany
{ddarma,v.petukhova,dietrich.klakow}@lsv.uni-saarland.de

Abstract

The paper presents an approach to answer extraction for a Question Answering Dialogue System (QADS), which is a part of an interactive quiz game. The information that forms the content of this game is concerned with biographical facts of famous people's life. The facts are extracted from Wikipedia pages by means of semantic relations, whose fillers are identified by trained sequence classifiers and pattern matching tools, and edited to be returned to the player as full-fledged system answers. The overall average F-score of 0.66 has been achieved, where for separate semantic relations F-score ranges from 0.21 to 0.90. The reported results show that the presented approach fits the data well and can be considered as a promising method for other QA domains, in particular when dealing with unstructured information.

1. Introduction

Question-Answering (QA) applications have gained steady growing attention over past decades. Three major approaches can be observed. The first one is the Information-Retrieval (IR) based QA system consisting of three main components: question processing, passage retrieval, and answer ranking (Moldovan et al., 2000). The second paradigm is a knowledge-based QA system as used by Apple Siri[1] and Wolfram Alpha [2]. Such systems, first, build a query representation and then map it to structured data like ontologies, gazeteers, etc. The third approach combines these two methods.

We aim at building an end-to-end Question Answering Dialogue System (QADS) that provides an interactive guessing game where players have to ask questions about attributes of an unknown person in order to guess his/her identity. The system adopts a statistical approach by employing the state-of-the-art machine-learning algorithms run on features such as n-grams, POS (Part-of-Speech), Named Entity (NE), syntactic chunks, etc. The main differences between our QA system and those of others, in general, is that our domain is rather closed, and the content that the system operates on is mainly unstructured free texts, however some databases available, e.g. Freebase[3]. What is more important, our system is an interactive QADS where the answers are returned to the user not as extracted information chunks or slot fillers, but are rather full-fledged dialogue utterances.

The core module of the QADS is the Dialogue Engine which consists of four main components such as interpretation module, dialogue manager, answer extraction module and utterance generation module[4]. The dialogue manager (DM) takes care of the overall communication between the user and the system. It gets as input from the interpretation module a dialogue act representation. Mostly it is about a question which is uttered by the human player. Questions are classified according to their communicative function (e.g. Propositional, Check, Set and Choice Questions) and semantic content. Semantic content is determined based on Expected Answer Type (EAT), e.g. LOCATION, and the focus word, e.g. *study*, see (Chernov et al., 2015). To extract the requested information, 59 semantic relations were defined that cover most important facts in human life, e.g. birth, marriage, career, etc. The extracted information is mapped to the EAT and focus word, and the most relevant answer and the strategy how to continue the dialogue are computed, see (Petukhova et al., 2015) for the later. DM then passes the system response for generation, where the DM input is transformed into a dialogue utterance (possibly multimodal one).

Designing the answer extraction module we set three objectives: (1) collection of unstructured data to create a dataset; (2) definition of the semantic relations and data annotation; (3) system design based on trained classifiers and post-processing tools to extract semantic relation automatically with reasonably high accuracy.

The paper is structured as follows. Section 2 gives an overview of previous approaches to QA system design. Section 3 defines semantic relations as a framework for this study. Section 4 describes the annotated data. In Section 5 the answer extraction procedure is depicted. Training experiments, evaluation results and their analysis can be found in Section6. Section 7 concludes the reported study and outlines future research.

2. Question Answering: related work

A breakthrough in QA has been made by (Moldovan et al., 2000) when designing an end-to-end open-domain QA system. This system achieved the best result in the TREC-8 competition[5] with accuracy of 77.7%. The system consists of three modules such as question processing, paragraph indexing and answer processing. First, the question type, question focus, question keyword and expected answer type are specified. Further, the search engine is used to retrieve the relevant documents and filter candidate paragraphs. Subsequently, the answer processing module identifies the answer in the paragraph using

[1] http://www.apple.com/ios/siri/

[2] www.wolframalpha.com

[3] http://www.freebase.com/

[4] Since Dialogue Engine is part of a larger distributed system which is the effort of an European consortium, ASR and TTS modules are not included in our local architecture and not discussed here.

[5] http://trec.nist.gov/pubs/trec8

lexico-semantic information (POS, Gazeteers, WordNet and Named Entities) and scoring candidates using word similarity metric and returns the answer with the highest scores.

In 2010, Watson a DeepQA system of IBM Research (Ferrucci et al., 2010) won a Jeopardy quiz challenge. This system incorporates content acquisition, question analysis, hypothesis generation, etc. Inside the hypotheses generation, it relies on named entity detection, triple store and reverse dictionary look-up to generate candidate answers which are then ranked based on confidence scores.

The most recent work comes from the TAC KBP slot filling task (Ellis, 2013) aiming to find filler(-s) for each identified empty slot, e.g. for a person (e.g. date_of_birth, age, etc.) and/or for a organization (e.g. member_of, founded_by, etc). Pattern matching, trained classifiers and Freebase[6] are used (Min et al., 2012; Roth et al., 2012) to find the best filler. The best system performance achieved in terms of F-score is 37.28% (Surdeanu, 2013) and (Roth et al., 2013).

TAC KBP approach differs from TREC tasks in that the former focuses on entities such as person or organization, while the later has broader focus (person, organization, location, etc). Secondly, TAC KBP slot filling has determined 41 slots that need to be filled, while in TREC, the information that needs to be found is dependent to the question. Finally, in terms of questions, TAC questions are defined by a topic and a list of slots that needs to be filled, while in TREC they vary from simple factoid to more complex questions.

Analysing the above mentioned studies, we concluded that computing an expected answer type (EAT), classification, pattern matching and named entity detection are important steps to robust answer extraction. Since our task, domain and data differ as mentioned above, the following extensions were performed:

- the TAC KBP 2013 relations set to compute EAT was enriched;
- three different Named Entity Recognizers (NERs) for better coverage of different types of NEs were applied;
- the matching patterns to capture the defined relations were designed;
- two sequence classifiers in three different settings were trained to better determine the exact answer's boundaries;
- ranked answer candidates were post-processed and redundancies removed before returning to the user.

3. Semantic framework: relations

To find a correct answer to a question semantic roles are often used. A semantic role is a relational notion describing the way a participant is involved in an event or state (Jackendoff, 1990) , typically providing answers to questions such as ”who” did ”what” to ”whom,” and ”when,” ”where,” ”why,” and ”how”. Along with semantic roles, relations between participants are also relevant for our domain, e.g. the relation between Agent and Co-Agent involved in ‘work’ event may be a COLLEAGUE_OF relation.

In order to decide on the set of relations to investigate, we collected game data in *Wizard of Oz* experiments, where one participant was acting as a Wizard simulating the system's behaviour (2 English native speakers: male and female) and the other as a game player (21 unique subjects: undergraduates of age between 19 and 25, who are expected to be related to our ultimate target audience). 338 dialogues were collected of total duration of 16 hours comprising about 6.000 speaking turns, see (Petukhova et al., 2014).

The experiments showed that most players tend to ask comparable questions about gender, place and time of birth or death, profession, achievements, etc. To capture this information we defined 59 semantic relations, from which 17 have been adopted from the TAC KBP 2013 Slot Filling task. TAC relations are mainly defined between NEs (persons and organizations), while our proposed set incorporates temporal event markers like TIME; captures PURPOSE and CAUSE relations between events; and introduces event modifiers like the MANNER marker; includes some domain-specific relations between entities such as AWARD, CREATOR_OF, COLLEAGUE_OF, OWNER_OF, etc. Moreover, we are not restricted to relations between NEs.

Each relation has two arguments and is one of the following types:

- RELATION(Z,?X), where Z is the person in question and X the entity slot to be filled, e.g. CHILD_OF(einstein,?X);
- RELATION(E_1, ?E_2) where E_1 is the event in question and E_2 is the event slot to be filled, e.g. REASON(death,?E_2); and
- RELATION(E,?X) where E is the event in question and X the entity slot to be filled, e.g. DURATION(study,?X).

The slots are primarily categorized based on the type of entities which we seek to extract information about. However, slots are also categorized by the content and quantity of their fillers (Ellis, 2013).

Slots are labelled as *name*, *value*, or *string* based on the content of their fillers. *Name* slots are required to be filled by the name of a person, organization, or geo-political entity (GPE). *Value* slots are required to be filled by either a numerical value or a date. The numbers and dates in these fillers can be spelled out, e.g. *December 7, 1941*, or written as numbers, e.g. *42* or *12/7/1941*. *String* slots are basically a ”catch all”, meaning that their fillers cannot be neatly classified as names or values.

Slots can be as *single-value* or *list-value* based on the number of fillers they can take. While single-value slots can have only a single filler, e.g. date of birth, list-value slots can take multiple fillers as they are likely to have more than one correct answer, e.g. employers.

4. Data

The data has been collected from Wikipedia[7]. 100 person's descriptions in English have been selected containing

[6] http://www.freebase.com/

[7] www.wikipedia.org

RELATION	%	RELATION	%	RELATION	%	RELATION	%	RELATION	%
ACCOMPLISHMENT	4.0%	DURATION	1.8%	LOC_DEATH†	0.8%	PART_IN	3.6%	TIME	14.6%
AGE_OF†	2.1%	EDUCATION_OF†	4.2%	LOC_RESIDENCE†	3.2%	RELIGION†	0.7%	TIME_BIRTH†	2.8%
AWARD	2.5%	EMPLOYEE_OF†	2.2%	MEMBER_OF†	1.8%	SIBLING_OF†	2.3%	TIME_DEATH†	1.0%
CHILD_OF†	3.6%	FOUNDER_OF†	1.2%	NATIONALITY†	3.1%	SPOUSE_OF†	1.9%	TITLE†	14.2%
COLLEAGUE_OF	1.7%	LOC	5.6%	OWNER_OF	1.1%	SUBORDINATE_OF	1.3%		
CREATOR_OF	8.5%	LOC_BIRTH†	5.0%	PARENT_OF†	3.7%	SUPPORTEE_OF	1.1%		

Table 1: List of defined semantic relations. † means that the relation is adopted from TAC KBP slot filling task.

1616 sentences (16 words/sentence on average), 30.590 tokens (5.817 unique tokens).

4.1. Data annotation and encoding

Descriptions are annotated using complex labels consisting of an IOB-prefix (**I**nside, **O**utside, and **B**eginning) and relation tag. We mainly focus on labeling nouns and noun phrases. For example:

(1) *Gates graduated from* ***Lakeside School*** *in 1973.*

The word *Lakeside* in (1) is labeled as the beginning of an EDUCATION_OF relation (B-EDUCATION_OF), and *school* is marked as inside of the label (I-EDUCATION_OF).

To assess the usability and reliability of the defined tagset, the inter-annotator agreement was measured in terms of the standard Kappa statistic (Cohen, 1960). For this, 10 randomly selected descriptions were annotated by two trained annotators. The obtained *kappa* scores were interpreted as annotators having reached good agreement (averaged for all labels, kappa = .76).

Table 1 gives an overview of the most frequently occurring relations in our data. In total, 3988 relations were identified, where TITLE was the most frequent one (562 entities) and LOC_DEATH the least frequent one (32 entities).

5. Answer extraction

Figure 1 depicts the answer extraction procedure. The process starts with splitting the data into training and test sets, 80% and 20% respectively. Subsequently, features are extracted for both sets and two sequence classifiers are applied. Additionally, a pattern matching tool is used to predict the outcome based on regular expressions. All predictions are then post-processed to return the final answer.

5.1. Classifiers, features and evaluation

Two well-known sequence classifiers such as Conditional Random Field (CRF) (Lafferty et al., 2001) and Support Vector Machine (SVM) (Joachims et al., 2009) are trained.[8]

The selected set of features includes **word & lemma tokens** as two basic features for classifiers; **POS** tags from the Stanford POS tagger (Toutanova et al., 2003); **NER** tags from three different NER tools: Stanford NER (Finkel et al., 2005), Illinois NER (Ratinov and Roth, 2009), and Saarland NER (Chrupala and Klakow, 2010); **chunking** using OpenNLP [10] to determine the NP boundaries; **key word** to determine the best sentence candidate for a particular relation, e.g. *marry, married, marriage, husband, wife, widow, spouse* for the SPOUSE_OF relation; **capitalization** to detect relations between NEs.

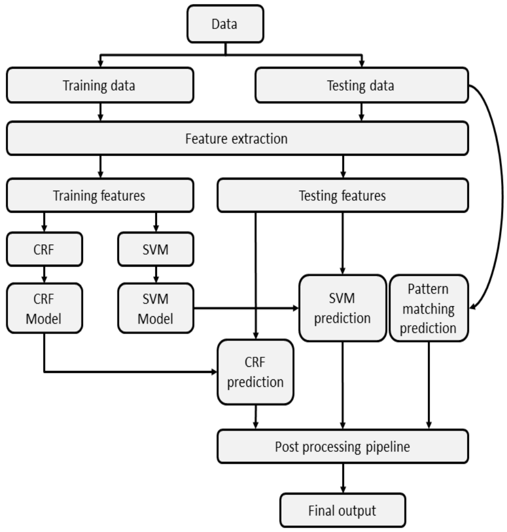

Figure 1: Answer extraction pipeline.

To assess the system performance standard evaluation metrics are used, precision (P), recall (R) and F-score (F1), using the tool developed by (Tjong Kim Sang and Buchholz, 2000). In particular, precision is important, since it is worse for the game to give the wrong answer than to say it cannot answer a question.[11] A classifier prediction is considered as correct if both the IOB-prefix and the relation tag fully correspond to those in the referenced annotation.

5.2. Pattern matching

Our pattern matching system handles 12 relations (See Table 4). These manually defined regular expressions seem to work well with certain relations. For example, regular expression like `born in (.*)` would match TIME_BIRTH or LOC_BIRTH relations. Subsequently, NER disambiguates between a DATE or GPE entities.

5.3. Post-processing procedures

The process of extracting relations does not stop after the classifiers and pattern matching tools are applied. Certain post-processing is required in order to select the best

[8] We used two CRF implementations from CRF++[9] and CRFsuite (Okazaki, 2007) with Averaged Perceptron (AP) and Limited-memory BFGS (L-BFGS) training methods.

[10] http://opennlp.apache.org/

[11] WoZ experiments participants indicated that 'not-providing' an answer was entertaining, giving wrong information, by contrast, was experienced as annoying.

	Baseline			System 1			System 2			System 3		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
CRF ++	0.56	0.34	0.42	0.68	0.52	0.59	0.82	0.55	0.66	0.85	0.54	0.66
CRFs_AP	0.33	0.29	0.30	0.54	0.53	0.53	0.71	0.57	0.63	0.74	0.56	0.64
CRFs_LBFGS	0.37	0.65	0.44	0.67	0.52	0.58	0.82	0.53	0.65	0.85	0.53	0.65
SVM-HMM	0.59	0.28	0.38	0.53	0.51	0.52	0.72	0.47	0.57	0.75	0.47	0.58
Pattern*	-	-	-	-	-	-	0.74	0.62	0.67	0.77	0.63	0.69

Table 2: Overall system performance. *) applied only to 12 most frequently occurred relations

Relation	P	R	F1	Relation	P	R	F1
ACCOMPLISHMENT	0.73	0.44	0.55	NATIONALITY	0.92	0.73	0.81
AGE_OF	0.95	0.76	0.84	OWNER_OF	0.76	0.40	0.48
AWARD	0.80	0.62	0.70	PARENT_OF	0.79	0.54	0.63
CHILD_OF	0.74	0.58	0.65	PART_IN	0.25	0.05	0.08
COLLEAGUE_OF	0.78	0.32	0.43	RELIGION	0.60	0.16	0.24
CREATOR_OF	0.64	0.17	0.26	SIBLING_OF	0.92	0.69	0.78
DURATION	0.97	0.64	0.76	SPOUSE_OF	0.76	0.42	0.52
EDUCATION_OF	0.84	0.65	0.72	SUBORDINATE_OF	0.81	0.19	0.31
EMPLOYEE_OF	0.77	0.19	0.28	SUPPORTEE_OF	1.00	0.40	0.54
FOUNDER_OF	0.65	0.26	0.36	MEMBER_OF	0.65	0.14	0.21
LOC	0.77	0.33	0.45	TIME	0.90	0.83	0.86
LOC_BIRTH	0.94	0.84	0.89	TIME_BIRTH	0.92	0.89	0.90
LOC_DEATH	0.90	0.55	0.67	TIME_DEATH	0.94	0.79	0.86
LOC_RESIDENCE	0.86	0.55	0.66	TITLE	0.84	0.66	0.74

Table 3: CRF++ performance on System 3.

Relation	P	R	F1	Relation	P	R	F1
AGE_OF	0.85	0.79	0.82	MEMBER_OF	0.46	0.43	0.42
CHILD_OF	0.87	0.87	0.87	PARENT_OF	0.86	0.78	0.82
DURATION	0.90	0.68	0.77	SIBLING_OF	0.93	0.85	0.88
EMPLOYEE_OF	0.53	0.16	0.23	SPOUSE_OF	0.79	0.63	0.70
FOUNDER_OF	0.74	0.71	0.72	SUBORDINATE_OF	0.72	0.61	0.65
LOC_DEATH	0.40	0.23	0.28	TIME_DEATH	0.29	0.23	0.26

Table 4: Pattern matching performance.

result for each relation, e.g. based on confidence scores. This step also involves eliminating relations that do not link the person in question and chunk expansion.

Relations that are not concerned with the person in question were removed. For example:

(2) *Her mother, Kathy Hilton is* ***a former actress****, and her father, Richard Howard Hilton, is* ***a businessman****.*

In (2), the classifier marks *a former actress* and *a businessman* as the TITLE. However, this relation does not link the person in question, but her mother and father. In other words, we omitted the TITLE relation from the same sentence that contains CHILD_OF and PARENT_OF relations.

There is also a special treatment for the TITLE relation which often requires chunk expansion when more information in form of complex possessive constructions is available. For example:

(3) *She later became* ***managing director of info service****.*

The output from our classifier for (3) has *managing director* as TITLE, while the correct chunk is *managing director of info service*. Therefore, we expand the relevant chunk in order to cover the full NP with embedded NPs inside.

6. Experimental setup and results

In our 5-fold cross-validation classification experiments, classifiers were trained and evaluated in three different settings: (1) *System 1* where classification is based on automatically derived features such as n-grams for word and lemma (trigrams), POS, NER tags, chunking and capitalization; the joint classification on all relations was performed; (2) *System 2*: pattern matching and classification on the same features as System 1 applied for each relation separately; and (3) *System 3*: the post-processed output of *System 2*.

All systems show the gains over the baseline systems. The later is obtained when training classifiers on word token features only. To indicate how good statistical classifiers generally are on relation recognition, consider the performance of distant supervision SVM[12] with precision of 53.3, recall of 21.8 and F-score of 30.9 (see (Roth et al., 2013)) on the TAC KBP relations. However, we emphasize that our task, relation set, application and data are different from those of TAC KBP. It would be useful in the future to test how well our proposed systems would behave on a different dataset.

As it can be observed from Table 2, the CRF++ classifier achieves the best results in terms of precision and F-score. Although the running time was not measured, the classification runs faster comparing to SVM-HMM. System 2 outperforms the System 1 (6-11% increase in F-score). When training on each relation in isolation, features weights can be adjusted more efficiently not affecting other relations classification. Moreover, this allows assigning multiple relations to the same entity more accurately while avoiding high data sparseness opposed to training on complex multi-classs labels. Key word features have been observed as having the highest information gain. Pattern matching is proven to be a powerful and straightforward

[12] Distant supervision method is used when no labeled data is available, see (Mintz et al., 2009).

method, see Table 4.

While in general System 3 gains a small increase in F-score (around 0.6-2%) compared to System 2, it increases the precision for many relations. More detailed results from CRF++ on System 3 can be seen in Table 3.

7. Conclusions and future work

We have discussed an approach for answer extraction from unstructured textual data. Our results showed that when dealing with each relation in isolation, better results can be achieved. Most of the relations can be identified correctly by training CRF++ sequence classifiers. Using pattern matching in addition to the classifiers can boost the performance of the whole system. Post-processing is required to refine the final output.

There is a lot of room for further research and development. From our observation, some of the relations are found using classification tools and not with pattern matching (and vice versa). In the future, both techniques should be combined. Observed inter-annotator agreement indicated that some relations need to be re-defined. Adding more training instances is expected to have a positive impact on the system's performance. In order to get a better coverage for the key words that appeared a very useful feature, synset information from WordNet[13] will be used. Finally, we will test how generic the proposed approach is by testing it on the TAC and TREC datasets.

8. References

Chernov, V., V. Petukhova, and D. Klakow, 2015. Linguistically motivated question classification. In *Proceedings of the 20th Nordic Conference on Computational Linguistics (NODALIDA)*. Vilnus, Lithuania.

Chrupala, G. and D. Klakow, 2010. A named entity labeler for german: Exploiting wikipedia and distributional clusters. In *Proceedings of LREC'10*. Valletta, Malta: European Language Resources Association (ELRA).

Cohen, J., 1960. A coefficient of agreement for nominal scales. *Education and Psychological Measurement*, 20:37–46.

Ellis, J., 2013. TAC KBP 2013 slot descriptions.

Ferrucci, D., E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. Kalyanpur, A. Lally, J. Murdock, E. Nyberg, J. Prager, N. Schlaefer, and C. Welty, 2010. Building watson: An overview of the deepqa project. *AI Magazine*, 31(3):59–79.

Finkel, J., T. Grenager, and C. Manning, 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of ACL '05*. Stroudsburg, PA, USA: Association for Computational Linguistics.

Jackendoff, R.S., 1990. *Semantic structures*. MIT Press.

Joachims, T., T. Finley, and C. Yu, 2009. Cutting-plane training of structural svms. *Machine Learning*, 77(1):27–59.

Lafferty, J., A. McCallum, and F. Pereira, 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML '01*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Min, B., X. Li, R. Grishman, and S. Ang, 2012. New york university 2012 system for kbp slot filling. In *Proceedings of the 5th Text Analysis Conference (TAC 2012)*.

Mintz, M., R. Bills, S.and Snow, and Jurafsky D., 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint ACL/IJCNLP Conference*.

Moldovan, D., S. Harabagiu, M. Pasca, R. Mihalcea, R. Girju, R. Goodrum, and V. Rus, 2000. The structure and performance of an open-domain question answering system. In *Proceedings of ACL '00*. Stroudsburg, PA, USA: Association for Computational Linguistics.

Okazaki, N., 2007. Crfsuite: a fast implementation of conditional random fields (crfs).

Petukhova, V., H. Bunt, A. Manchanau, and R. Aruchamy, 2015. Experimenting with grounding strategies in dialogue. In *Proceedings of the GoDial 2015 Workshop on the Semantics and Pragmatics of Dialogue*. Goteborg, Sweden.

Petukhova, V., M. Gropp, D. Klakow, G. Eigner, M. Topf, S. Srb, P. Moticek, B. Potard, J. Dines, O. Deroo, R. Egeler, U. Meinz, and S. Liersch, 2014. The DBOX corpus collection of spoken human-human and human-machine dialogues. In *Proceedings of the 9th Language Resources and Evaluation Conference (LREC)*. ELDA, Reykjavik, Iceland.

Ratinov, L. and D. Roth, 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of CoNLL '09*. Stroudsburg, PA, USA: Association for Computational Linguistics.

Roth, B., T. Barth, M. Wiegand, M. Singh, and D. Klakow, 2013. Effective slot filling based on shallow distant supervision methods. In *TAC KBP 2013 Workshop*. Gaithersburg, Maryland USA: National Institute of Standards and Technology.

Roth, B., G. Chrupala, M. Wiegand, M. Singh, and D. Klakow, 2012. Saarland university spoken language systems at the slot filling task of tac kbp 2012. In *Proceedings of the 5th Text Analysis Conference (TAC 2012)*. Gaithersburg, Maryland, USA.

Surdeanu, M., 2013. Overview of the tac2013 knowledge base population evaluation: English slot filling and temporal slot filling. In *TAC KBP 2013 Workshop*. Gaithersburg, Maryland USA: National Institute of Standards and Technology.

Tjong Kim Sang, E. and S. Buchholz, 2000. Introduction to the conll-2000 shared task: chunking. In *Proceedings of the 2nd workshop on Learning Language in Logic and ConLL '00*. Stroudsburg, PA, USA: Association for Computational Linguistics.

Toutanova, K., D. Klein, C. Manning, and Y. Singer, 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of NAACL '03*. Stroudsburg, PA, USA: Association for Computational Linguistics.

[13] http://wordnet.princeton.edu/