Understanding Questions and Extracting Answers: Interactive Quiz Game Application Design*

Volha Petukhova, Desmond Darma Putra, Alexandr Chernov and Dietrich Klakow
Spoken Language Systems Group, Saarland University, Saarbrücken, Germany
{v.petukhova,a.chernov,d.darma,d.klakow}@lsv.uni-saarland.de

October 6, 2019

Abstract

The paper discusses two key tasks performed by a Question Answering Dialogue System (QADS): user question interpretation and answer extraction. The system represents an interactive quiz game application. The information that forms the content of the game is concerned with biographical facts of famous people's life. The process of a question classification and answer extraction is performed based on a domain-specific taxonomy of semantic roles and relations computing the Expected Answer Type (EAT). Question interpretation is achieved performing a sequence of classification, information extraction, query formalization and query expansion tasks. The expanded query facilitates the search and retrieval of the information. The facts are extracted from Wikipedia pages by means of the same set 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 results (precision of 85% for the EAT classification of both in questions and answers) 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 [5]. The second paradigm is a knowledge-based QA system as used by Apple Siri[1], Amazon Alexa[2], Wolfram Alpha [3], etc. Such systems, first, build a query representation and then map it to structured data like ontologies, gazetteers, 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 state-of-the-art supervised 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 the others, in general, are that our domain is rather closed, and that the content that the system operates on is mainly unstructured free texts. 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: interpretation module, dialogue manager, answer extraction module and utterance generation module. The Dialogue Manager (DM) takes care of the overall communication between the user and the system. It gets as an 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 identifying their communicative function (e.g. Propositional, Check, Set and Choice Questions) and semantic content in accordance to the ISO 24617-2 dialogue act standard [20]. Semantic content is determined

*This is a pre-print version of the following paper: Volha Petukhova, Desmond Darma Putra, Alexandr Chernov and Dietrich Klakow (2018) Understanding Questions and Extracting Answers: Interactive Quiz Game Application Design Human Language Technology. Challenges for Computer Science and Linguistics. Zygmunt Vetulani and Joseph Mariani (eds.), Lecture Notes in Computer Science, Springer, Berlin

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

[2] https://developer.amazon.com/alexa

[3] www.wolframalpha.com

based on Expected Answer Type (EAT). To extract the requested information, a taxonomy is designed comprising 59 semantic relations to cover the most important facts in human life, e.g. birth, marriage, career, etc. The extracted information is mapped to the EAT, and both the most relevant answer and a strategy for continuing the dialogue are computed. The Dialogue Manager then passes the extracted information for further system response generation, where the DM input is transformed into a dialogue utterance (possibly multimodal one).

For a closed domain as ours, restricted to personal biographical facts, it is possible to narrow down the knowledge available to the system. For example, structured knowledge bases can be used, e.g. Freebase[4]. They are, however, not complete to achieve sufficient coverage of factual information required for our game. Therefore, the content that the system operates on is a bigger collection of unstructured free texts, namely Wikipedia articles[5]. This impacts search and retrieval tasks. As a consequence, the output of the question understanding module should be a rather comprehensive query capturing various semantic information concerning events in question, entities involved in this event and their properties, and type of relations between entities and possibly between events, EAT. The EAT is augmented with question focus word(-s) to determine the main event in question. The EAT together with focus word(-s), are formalized in a query which, on its turn, is expanded to cover as many natural language variations as possible. To extract the requested information, information is mapped to the EAT and focus word. The answer extraction module operates on unstructured unprocessed data as input, i.e. Wikipedia articles, and its 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. Classification results using semantic relations in questions (Section 5) and for answer extraction (Section 6) are presented. We also outline performed experiments describing features, algorithms and evaluation metrics that have been used. We discuss how the query is generated and expanded, and how the full answer extraction procedure is designed. Section 6 concludes the reported study and outlines future research.

2 Question Answering: related work

A breakthrough in QA has been made by [5] when designing an end-to-end open-domain QA system. This system achieved the best result in the TREC-8 competition[6] 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 lexico-semantic information (POS, Gazetteers, WordNet and Named Entities), and after scoring candidates using word similarity metric returns the highest ranked answer.

In 2010, Watson, a DeepQA system of IBM Research [3], won a Jeopardy quiz challenge. This system incorporates content acquisition, question analysis, hypothesis generation, etc. For 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 [8] aimed at finding fillers for each identified empty slot, e.g. for 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 are used in [1, 2] to find the best filler. The best system performance achieved in terms of F-score is 37.28% [21] and [23].

The TAC KBP approach differs from TREC tasks in that the former focuses on entities such as person or organization, while the later has a 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 dependes on the question asked. 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 factoids to more complex questions.

Analysing the above mentioned studies, we concluded that computing the Expected Answer Type (EAT), classification, focus word extraction, query generation and expansion and pattern matching are important steps to robust question classification and answer extraction. Since our task, domain and data differ as mentioned above, the following extensions were performed:

- the TAC KBP 2013 relations set was enriched to compute EAT;

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

[5] http://www.wikipedia.org

[6] `http://trec.nist.gov/pubs/trec8`

- different syntactic and semantic parsers for better coverage of relevant phenomena were applied;
- different classifiers and classification procedures were evaluated to determine the EAT in questions and answer candidates, and to establish the answer's boundaries;
- the EAT information was enriched with query focus word(-s) and expanded with synonyms to enable efficient and accurate answer extraction;
- matching patterns to capture the defined relations were designed;
- ranked answer candidates were post-processed and redundancies removed.

3 Semantic framework: relations

To understand a question and to find a correct answer to this 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 [19], typically providing answers to questions such as"who" did "what" to "whom", and "when", "where", "why", and "how". Several semantic role annotation schemes have been developed in the past, e.g. FrameNet [33], PropBank [32] and Lirics [34]. Along with semantic roles, relations between participants are also relevant for our domain, e.g. the relation between Agent and Co-Agent (or Partner) involved in a 'work' event may be a COLLEAGUE_OF relation.

Depending on the domain and task, QA systems may require different kinds of question and answer type taxonomies. For instance, Singhal et al. (2000) designed a very simple taxonomy based on the correspondence between question words and expected answer types. For instance, according to this taxonomy, questions containing *who* or *whom* answers of the type *person*. For more ambiguous question words like *what* or *which* the type of a question was identified by the head noun.

Moldovan et al. (2000) define 9 question classes (e.g. *'what'*, *'who'*, *'how'*) and 20 sub-classes (e.g. *'basic what'*, *'what-who'*, *'what-when'*). Additionally, expected answer type is determined, e.g. *person*, *money*, *organization*, *location*. Finally, a focus word or a sequence of words is identified in the question, which disambiguates it by indicating what the question is looking for (see [5] for an overview of defined classes for 200 of the most frequent TREC-8 questions).

Li and Roth (2002) proposed another question classification scheme, also based on determining the expected answer type. This scheme is a layered hierarchical two-level taxonomy. The first level represents coarse classes like *Date*, *Location*, *Person*, *Time*, *Definition*, *Manner*, *Number*, *Title*, *Organization*, *Reason* , etc. The second level comprises 50 fine-grained classes like *Description*, *Group*, *Individual* and *Title* for the upper-level class of *Human*. Using a hierarchical classifier they tried to get an increase in performance, but experimental results showed that the gained difference with a flat classifier was not statistically significant.

The TAC KBP slot filling task [8] aimed at finding fillers for each identified empty slot, e.g. for a person (e.g. date_of_birth, age, etc.) and/or for an organization (e.g. member_of, founded_by, etc).

To decide on the set of relations to investigate, we analysed already available and collected new dialogue data. As a starting point, we analysed recordings of the famous US game 'What's my line?' that are freely available on Youtube[7]. However, the latter differs from our scenario: during the TV-show participants may ask only propositional questions with expected 'yes' or 'no' answers; our game allows any question type from the user. Therefore, we collected data in pilot dialogue experiments, where one participant was acting as a person whose name should be guessed and the other as a game player. 18 dialogues were collected of total duration of 55 minutes comprising 360 system's and user's speaking turns. To evaluate the relation set and to train classifiers, we performed large scale gaming experiments in a Wizard of Oz setting, see next section.

Pilot experiments showed that all players tend to ask similar questions about gender, place and time of birth or death, profession, achievements, etc. To capture this information we defined 59 semantic relations proposing a multi-layered taxonomy: a high level, coarse annotation comprising 7 classes and a low-level, fine-grained annotation, comprising 52 classes, see [27] for more details. This includes the HUMAN DESCRIPTION class defined for acts about an individual such as age, title, nationality, etc.; HUMAN RELATIONS for family relations; HUMAN GROUPS for relations between colleagues, friends, etc.; EVENTS & NON-HUMAN ENTITIES class for awards, products of human activities, etc.; EVENT MODIFIERS for specifying manner, reasons, etc.; the TIME class to capture temporal information like duration, frequency, etc.; and the LOCATION class to capture spatial event markers for places where events occur. Table 1 presents the subset of about the 30 most frequently occurring relations with an indication of

[7] https://www.youtube.com/channel/UChPE75Fvvl1HmdAsO7Nzb8w

RELATION	Questions (%)	Answers (%)	RELATION	Questions (%)	Answers (%)
ACTIVITY_OF	10.2	4.0	LOC_BIRTH	2.3	5.0
AGE_OF① ②	3.0	2.1	AWARD	4.4	2.5
LOC_RESIDENCE	1.7	3.2	MEMBER_OF①	2.4	1.8
CHILD_OF①	1.5	3.6	COLLEAGUE_OF	1.0	1.7
NATIONALITY①	1.2	3.1	CREATOR_OF	6.1	8.5
OWNER_OF	2.0	1.1	PARENT_OF①	1.3	3.7
DURATION④	1.3	1.8	EDUCATION_OF①	3.7	4.2
RELIGION	2.5	0.7	EMPLOYEE_OF①	1.6	2.2
SIBLING_OF①	0.9	2.3	SPOUSE_OF①	1.4	1.9
FAMILY_OF	1.6	-	FOUNDER_OF①	1.9	1.2
TIME② ③ ④	8.0	14.6	TIME_BIRTH	2.1	2.8
TIME_DEATH	1.6	1.0	LOCATION ② ③ ④	4.7	5.6
TITLE① ③	11.1	14.2	LOC_DEATH	1.7	0.8
PART_IN	-	3.6	CHARGED_FOR	4.2	-
GENDER	1.7	-	NAME	1.9	-

Table 1: Question and answer types in terms of defined semantic relations and their distribution in data (relative frequency in %; (① means that the relation is also defined in TAC KBP slot filling task; ② in TREC-08 QA task; ③ in TREC 2002 QA task, i.e. annotation scheme proposed by [10]; and ④ in LIRICS semantic role set)).

what concepts can be found in existing schemes for annotating semantic relations and semantic roles. We also provide relative frequencies of the annotated questions and answers in the data. It should be noted here that the majority of the concepts defined here are domain-specific, i.e. tailored to our quiz game application. The approach could however be adapted for designing comparable annotation schemes for other domains; this has for example been done for the food domain (see [35]).

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 categorized by the entity type which we seek to extract information about. However, slots are also categorized by the content and quantity of their fillers [8].

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, e.g. *December 7, 1941*, *42*, *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 having more than one correct answer, e.g. employers.

4 Data: collection and annotations

In order to validate the proposed EAT annotation scheme empirically and to build an end-to-end QADS, two types of data are required: (1) dialogue data containing player's questions that are more realistic than youtube games and larger than our pilots; and (2) descriptions containing answers to player's questions about the guessed person.

To collect question data we explored different possibilities. There is some question data publicly available, e.g. approximately 5500 questions are provided by the University Illinois[8] annotated according to the scheme defined in [10]. However, not all of this data can be used for our scenario. We filtered out about 400 questions for our purposes. Since this dataset is obviously too small, we generated questions automatically using the tool provided by (Heilman and Smith, 2009) from the selected Wikipedia articles and filtered them out manually: grammatically broken questions were fixed and repetitions deleted. Additionally, synonyms from WordNet[9] were used to generate different variations of questions for the same class.

[8] `http://cogcomp.cs.illinois.edu/page/resources/data`

[9] urlhttp://wordnet.princeton.edu/

We collected game data in *Wizard of Oz* experiments on a larger scale then pilot ones. Here again 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 represent our target audience). 338 dialogues were collected of total duration of 16 hours comprising about 6.000 speaking turns, see [26].

The final question set consists of 1069 questions. Table 1 illustrates the distribution of question and answer types in terms of the EAT.

Additionally, a focus word or words sequence specifying the main event in a question, usually a verb or eventive noun, is extracted from the question to compute the EAT and formulate the query. For example,

(1) Question: When was his first album released?
Assigned semantic relation: TIME
Focus word sequence: first album released
EAT: TIME_release(first_album)
Query: TIME_release(first_album) :: (E, ?X) :: QUALITY(VALUE) :: QUANTITY(SINGLE)

Answers were retrieved from 100 selected English Wikipedia articles containing 1616 sentences (16 words/sentence on average), 30.590 tokens (5.817 unique tokens). Descriptions are annotated using complex labels consisting of an IOB-prefix (**I**nside, **O**ut-
side, and **B**eginning) and the EAT tag to learn the exact answer boundaries. We mainly focus on labeling nouns and noun phrases. For example:

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

The word *Lakeside* in (2) 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). Table 1 illustrates the distribution of the most frequently occurring answer types based on the identified semantic relation.

Since the boundaries between semantic classes are not always clear, we allowed multiple class labels to be assigned to one entity. For example:

(3) *Living in Johannesburg, he became involved in anti-colonial politics, joining the ANC and becoming a founding member of its **Youth League**.*

Here, *Youth League* is founded by a person (FOUNDER_OF relation), but the person is also a member of the *Youth League*. There are also some overlapping segments detected as in example (4):

(4) *He served as **the commander-in-chief of the Continental Army** during the American Revolutionary War.*

The entity *commander-in-chief of the Continental Army* in (4) is marked as TITLE, while *the Continental Army* is recognized as MEMBER_OF. Both of these relations are correct, since if a person leads an army he/she is also a member of it.

To assess the reliability of the defined tagset, the inter-annotator agreement was measured in terms of the Cohen's kappa [22]. For this, 10 randomly selected descriptions and all 1069 questions were annotated by two trained annotators. The obtained *kappa* scores were interpreted as annotators having reached good agreement (averaged for all labels, kappa = .76).

5 Question classification

5.1 Classification design: classifiers, features and evaluation

We defined the question classification task as a machine-learning task, for which we built Support Vector Machines (SVM, [4]) based classifiers. In all experiments, linear kernel function (linearSVC) was used. We performed stratified 5-fold cross-validation multi-class and multi-label classification experiments applying cascade classification procedure. This implies that the first set of classifiers was trained to classify coarse labels (7 top classes) and coarse class predictions were added as features to perform fine-grained classification (cascading).[10]

We conducted a series of experiments to assess: (1) the features' importance for the defined task; and (2) classifiers performance in the cascade setting.

[10] Another classification procedure is known as *hierarchical* classification. Hierarchy of classifiers consists of classifier#1 deciding to which coarse class a question belongs and transfers this information to the corresponding classifier trained specifically to predict this particular question type.

Experiment 1						Experiment 2					
Features	**n-grams range**					**Features**	**n-grams range**				
	1,1	**1,2**	**2,2**	**2,3**	**3,3**		**1,1**	**1,2**	**2,2**	**2,3**	**3,3**
Words	0.81	0.81	0.72	0.72	0.68	+CP	0.82	0.81	0.77	0.76	0.70
POS	0.27	0.40	0.40	0.46	0.44	+CP	0.41	0.60	0.59	0.60	0.56
Lem	0.80	**0.82**	0.74	0.73	0.71	+CP	0.81	**0.82**	0.78	0.78	0.75
Words+POS	0.80	0.81	0.71	0.71	0.68	+CP	0.81	0.81	0.77	0.76	0.70
Lem+POS	0.80	0.82	0.73	0.73	0.71	+CP	0.81	0.82	0.78	0.78	0.75
Focus	0.75	0.74	0.63	0.59	0.31	+CP	0.79	0.77	0.68	0.65	0.44
FocusLem	0.76	0.76	0.65	0.61	0.39	+CP	0.79	0.79	0.71	0.68	0.48

Table 2: Precision of the question classifier for fine classes (CP - coarse class labels predicted by the classifier).

Since the system's goal is to provide the player with a correct answer, and if no answer was found to acknowledge the fact by generating negative feedback utterances like "*Sorry, I do not have this information*"[11], the classifier precision has been considered in evaluation. The trained classifiers performance was compared to the baseline. The baseline was computed based on a single feature, namely, *bag-of-words* when training the Naive Bayes classifier. The baseline classifier was implemented using Multinomial Naive Bayes algorithm from scikit-learn [29] achieving the precision of 56%.

Features computed from the data include *bag-of-words, bigrams, trigrams* and *part-of-speech (POS) tags* using the Stanford CoreNLP tools[12]. In our experiments *surface word forms* and *lemmas* of *focus words* as the most salient words were used as features. Apart from that, we applied combinations of all the above mentioned features with coarse class labels to predict fine classes.

To extract focus words, we implemented an algorithm that preserves the main nominal phrase with the predicate, corresponding prepositions and conjunctions while removing everything else. The algorithm excludes stop words and stop phrases (from predefined lists), as well as some parts of speech (based on the Penn Tree Bank tagset[13] we remove existential *there*, interjections, interrogative pronouns and possessive endings), auxiliary verbs, and interrogative pronouns. Questions from the real dialogue data were manually annotated with focus words, which allowed to test this algorithm. It was able to extract focus words with the accuracy of 94.6%.

5.2 Experimental results

As for features, the best results were obtained by the model trained on *unigrams + bigrams of lemmas*. In most cases models based on *unigrams+bigrams* demonstrated significantly better results than *unigram, bigram* or *trigram models*. It means that the word order is important to classify questions correctly. Our classifier outperformed the baseline (X^2 (1, n = 2403) = 293.181, p<.05). Table 2 summarizes results from all experiments.

Adding coarse class labels as additional features did not result in a significantly higher precision. The classifier that predicts coarse class labels achieved the average precision of 90% . However, it was not enough to make the predicted coarse class labels useful as features for the fine-grained classification.

As for separate classes, questions of the most prevailing classes were identified with the highest precision: TITLE - 85%, CREATOR_OF - 81%, NAME - 89%.

As expected, the classifier achieved the best results by using lexical clues, i.e. the presence or absence of certain words is a strong feature to determine to which class or classes a question belongs. Unfortunately, when a question contains words shared by questions belonging to different classes, it caused prediction errors. Extensive error analysis and learnability experiments were performed, see [25].

5.3 Query Generation and Expansion

Query generation is the last data processing operation that is performed by the question interpretation module. The query is generated according to the pre-defined set of rules. It captures the results of the question classification process as well as the extracted focus words and transfers this information to the next module.

The query generation processes, the semantic representation of its components in particularly, partially based on the Discourse Representation Theory (DRT) [30]. It incorporates semantic information that is necessary to find the

[11]To make the game more entertaining, the system can always play with strategies to turn a negative situation in a system's favour. For example, if no answer was found, the system may ask the player to ask another question claiming that the previous one was not eligible for whatever reasons or the answer to it would lead to quick game end, or alike.

[12]http://nlp.stanford.edu/downloads/corenlp.shtml

[13]http://www.cis.upenn.edu/∼treebank/

Question	What do you do as a job?
Focus words	do as job
Expanded focus	do [make, perform, cause, practice, act], as, job [activity, occupation, career, employment, position]
EAT	Title_do(do as job)
Query	(Z, E, ?X) :: Title_do(Z, doAs, ?job) :: QUALITY(String) :: QUANTITY(List) :: FOCUS(do as job)
Expanded query	(Z, E, ?X) :: Title_do(Z, doAs, ?job) :: QUALITY(String) :: QUANTITY(List) :: FOCUS(do [make, perform, practice, act], as, job [activity, occupation, career, employment, position])

Table 3: Example of an expanded query.

correct answer. Table 3 demonstrates an example of such a query.

In natural language the same message may have a number of realizations. So far, our QA system misses many answers when the answer is expressed by different lexical units. To solve this problem, we used WordNet[14] synonyms to elaborate the extracted question focus words.

6 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.

6.1 Classifiers, features and evaluation

Two well-known sequence classifiers such as Conditional Random Field (CRF) [6] and Support Vector Machine (SVM) [9] are trained.[15]

The selected set of features includes *word* and *lemma* tokens as two basic features for classifiers; *POS* tags from the Stanford POS tagger [16]; *NER* tags from three different NER tools: Stanford NER [15], Illinois NER [11], and Saarland NER [17]; *chunking* using OpenNLP [17] to determine the NP boundaries; *key word* to determine the best sentence candidate for a particular relation, e.g. `marry`, `marriage`, `husband`, `wife`, `spouse` for the SPOUSE_OF relation; *capitalization* to detect relations between NEs.

To assess the system performance standard evaluation metrics are used, precision (P), recall (R) and F-score (F1), using the tool developed by [14]. 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.[18] A classifier prediction is considered as correct if both the IOB-prefix and the relation tag fully correspond to those in the referenced annotation.

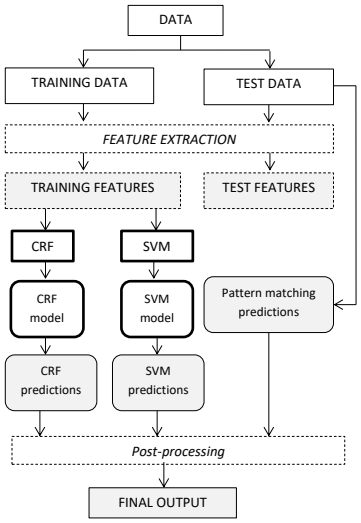

Figure 1: Answer extraction pipeline.

6.2 Pattern matching

Our pattern matching system handles 12 relations (See Table 6). These manually defined regular expressions seem to work well with certain relations. For example, regular expression like `born in (.*)` would match TIME_BIRTH

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

[15] We used two CRF implementations from CRF++[16] and CRFsuite [7] with Averaged Perceptron (AP) and Limited-memory BFGS (L-BFGS) training methods.

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

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

Classifier	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 4: Overall system performance. *) applied only to 12 most frequently occurred relations. P stands for precision; R for recall; F1 for harmonic mean.

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 5: CRF++ performance on System 3. P stands for precision; R for recall; F1 for harmonic mean.

or LOC_BIRTH relations. Subsequently, NER disambiguates between a DATE or GPE entities.

6.3 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 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:

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

In (5), 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:

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

The output from our classifier for (6) 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.

7 Experimental setup and results

In our 5-fold cross-validation classification experiments, classifiers were trained and evaluated in 3 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; (3) *System 3*: the post-processed output of *System 2*.

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 6: Pattern matching performance. P stands for precision; R for recall; F1 for harmonic mean.

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[19] with precision of 53.3, recall of 21.8 and F-score of 30.9 (see [23]) 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 4, 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 method, see Table 6.

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 5.

8 Conclusions and future work

We proposed a data-oriented approach for question classification and answer extraction from unstructured textual data based on determining semantic relations between entities computing the Expected Answer Type. Our results showed that the relations that we have defined help the system to understand user's questions and to capture the information, which needs to be extracted from the data.

Having analysed misclassified EATs, we drew several conclusions. First, the classifier confuses semantically similar classes. Second, the classifier has difficulty to identify EATs for under-represented classes. Third, questions simultaneously belonging to several classes were often misclassified.

The easiest way to achieve a higher precision is probably to increase the number of instances for the under-represented classes. Of course, it is impossible to force the users to ask only certain types of questions. However, new instances can be generated based on the designed patterns using bootstrapping techniques and user's behaviour simulations.

Some of the relations were 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. Finally, we will test how generic the proposed approach is by testing it on the TAC and TREC datasets. Moreover, since some relations, in particular of RELATION(E_1, ?E_2) and RELATION(E,?X) types, are very close to semantic roles, there is a need to analyse semantic role sets (e.g. ISO semantic roles [36]) and study the possible overlaps.

From the QADS development point of view, we need to evaluate the system in real settings. For this, question classifiers need to be re-trained on the actual and potentially erroneous ASR output. While testing/evaluating the QADS system, additional data will be produced, saved and used to enrich the training set.

9 Acknowledgments

The research reported in this paper was carried out within the DBOX Eureka project under number E! 7152.

[19] Distant supervision method is used when no labeled data is available, see [24].

References

[1] Min, B. and Li, X. and Grishman, R. and Ang, S.: New York University 2012 System for KBP Slot Filling In: Proceedings of the 5th Text Analysis Conference (TAC 2012) (2012)

[2] Roth, B., Chrupala, G., Wiegand, M., Singh, M. and Klakow, D.: 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 (2012)

[3] Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A., Lally, A., Murdock, J., Nyberg, E., Prager, J., Schlaefer, N. and Welty, C.: Building Watson: An Overview of the DeepQA Project. In: AI Magazine, 3(31), pp. 59-79. (2010)

[4] Cortes, C. and Vapnik, V.: Support Vector Networks. Machine Learning, 20 (3), pp. 273-297. (1995)

[5] Moldovan, D., Harabagiu, S., Pasca, M., Mihalcea, R., Girju, R., Goodrum, R. and Rus, V.: The structure and performance of an open-domain question answering system. In: Proceedings of the Association for Computational Linguistics, pp. 563–570 (2000)

[6] Lafferty, J., McCallum, A. and Pereira, F.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: Proceedings of ICML '01, pp. 282–289 (2001)

[7] Okazaki, N.: CRFsuite: a fast implementation of Conditional Random Fields (CRFs) http://www.chokkan.org/software/crfsuite/ (2007)

[8] Ellis, J.: TAC KBP 2013 Slot Descriptions. `http://surdeanu.info/kbp2013/TAC_2013_KBP_Slot_Descriptions_1.0.pdf` (2013)

[9] Joachims, T., Finley, T. and Yu, C.: Cutting-plane training of structural SVMs. Machine Learning, 77(1), pp. 27–59 (2009)

[10] Li, X. and Roth, D.: Learning question classifiers. In: Proceedings of the COLING '02, Association for Computational Linguistics, pp. 1–7 (2002)

[11] Ratinov, L. and Roth, D.: Design challenges and misconceptions in named entity recognition. In: Proceedings of CoNLL '09, Association for Computational Linguistics, pp. 147–155 (2009)

[12] Riloff, E. and Thelen, M.: A Rule-based Question Answering System for Reading Comprehension Tests. In: Proceedings of the 2000 ANLP/NAACL Workshop on Reading Comprehension Tests As Evaluation for Computer-based Language Understanding Sytems - Volume 6, pp.13–19 (2000)

[13] Singhal, A., Abney, S. P., Bacchiani, M., Collins, M., Hindle, D., and Pereira, F.: AT&T at TREC-8. In: TREC (2000)

[14] Tjong Kim Sang, E. and Buchholz, S.: Introduction to the CoNLL-2000 shared task: chunking. In: Proceedings of the 2nd workshop on Learning Language in Logic and ConLL '00, Association for Computational Linguistics, pp. 127–132 (2000)

[15] Finkel, J., Grenager, T. and Manning, C.: Incorporating non-local information into information extraction systems by Gibbs sampling. In: Proceedings of ACL '05, Association for Computational Linguistics, pp. 363–370 (2005)

[16] Toutanova, K., Klein, D., Manning, C. and Singer, Y.: Feature-rich part-of-speech tagging with a cyclic dependency network. In: Proceedings of NAACL '03, Association for Computational Linguistics, pp. 173–180 (2003)

[17] Chrupala,G. and Klakow, D.: A Named Entity Labeler for German: Exploiting Wikipedia and Distributional Clusters. In: Proceedings of LREC'10, European Language Resources Association (ELRA), pp. 552–556 (2010)

[18] Jackendoff, R.S.: Semantic interpretation in generative grammar. MIT Press, Cambridge (1972)

[19] Jackendoff, R.S.: Semantic structures. MIT Press, Cambridge (1990)

[20] ISO: Language resource management – Semantic annotation framework – Part 2: Dialogue acts. ISO 24617-2. ISO Central Secretariat, Geneva (2012)

[21] Surdeanu, M.: Overview of the TAC2013 Knowledge Base Population Evaluation: English Slot Filling and Temporal Slot Filling. In: Proceedings of the TAC KBP 2013 Workshop, National Institute of Standards and Technology (2013)

[22] Cohen, J.: A coefficient of agreement for nominal scales. Education and Psychological Measurement, 20, pp. 37-46 (1960)

[23] Roth, B., Barth, T., Wiegand, M., Singh, M. and Klakow, D.: Effective Slot Filling Based on Shallow Distant Supervision Methods. In: Proceedings of the TAC KBP 2013 Workshop, National Institute of Standards and Technology (2013)

[24] Mintz, M., Bills, S., Snow, R. and Jurafsky D.: Distant supervision for relation extraction without labeled data. In: Proceedings of the Joint ACL/IJCNLP Conference, pp. 1003-1011 (2009)

[25] Chernov, V., Petukhova, V. and Klakow, D.: Linguistically Motivated Question Classification. In: Proceedings of the 20th Nordic Conference on Computational Linguistics (NODALIDA), pp. 51–59 (2015)

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

[27] Petukhova, V.: Understanding questions and finding answers: semantic relation annotation to compute the expected answer type. In: Proceedings of the Tenth Joint ISO - ACL SIGSEM Workshop on Interoperable Semantic Annotation (ISA-10). Reykjavik, Iceland, pp. 44–52 (2014)

[28] Heilman, M.: Automatic Factual Question Generation from Text. PhD thesis, Carnegie Mellon University, USA (2011)

[29] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E.: Scikit-learn: Machine Learning in Python. In: Journal of Machine Learning Research, 12, pp. 2825–2830 (2011)

[30] Kamp, H. and Reyle, U.: From discourse to logic. Introduction to modeltheoretic semantics of natural language, formal logic and Discourse Representation Theory. In: Studies in Linguistics and Philosophy, 42, Kluwer, Dordrecht, The Netherlands (1993)

[31] Kipper, K.: VerbNet: A Class-Based Verb Lexicon. Available at `http://verbs.colorado.edu/~mpalmer/projects/verbnet.html` (2002)

[32] Palmer, M., Gildea, D., and Kingsbury, P.: The Proposition Bank: An Annotated Corpus of Semantic Roles. In: Computational Linguistics, 31(1), pp. 71-106 (2002)

[33] ICSI: FrameNet. Available at `http://framenet.icsi.berkeley.edu` (2005)

[34] Petukhova, V., and Bunt, H.: LIRICS semantic role annotation: Design and evaluation of a set of data categories. In: Proceedings of the sixth international conference on language resources and evaluation (LREC 2008), Paris: ELRA (2008)

[35] Wiegand, M. and Klakow, D.: Towards the Detection of Reliable Food-Health Relationships. In: Proceedings of the NAACL-Workshop on Language Analysis in Social Media (NAACL-LASM), pp. 69–79 (2013)

[36] Bunt, H., and Palmer, M.: Conceptual and Representational Choices in Defining an ISO standard for Semantic Role Annotation. In: Proceedings Ninth Joint ISO - ACL SIGSEM Workshop on Interoperable Semantic Annotation (ISA-9), Potsdam, pp. 41–50 (2013)