Master's Thesis

Submitted in Partial Fulfillment of the Requirements
for the Degree of
Master of Arts in Computational Linguistics

An Integrated Approach to Answer Selection in Question Answering: Exploring Multiple Information Sources and Domain Adaptation

Author:
Björn Rudzewitz

Supervisors:
Prof. Dr. Detmar Meurers
Prof. Dr. Fritz Hamm

Seminar für Sprachwissenschaft
Eberhard-Karls-Universität Tübingen

August 2016

Hiermit versichere ich, dass ich die Arbeit selbständig verfasst, keine anderen als die angegebenen Hilfsmittel und Quellen benutzt, alle wörtlich oder sinngemäß aus anderen Werken übernommenen Aussagen als solche gekennzeichnet habe und dass die Arbeit weder vollständig noch in wesentlichen Teilen Gegenstand eines anderen Prüfungsverfahrens gewesen ist und dass die Arbeit weder vollständig noch in wesentlichen Teilen bereits veröffentlicht wurde sowie dass das in Dateiform eingereichte Exemplar mit den eingereichten gebundenen Exemplaren übereinstimmt.

I hereby declare that this paper is the result of my own independent scholarly work. I have acknowledged all the other authors' ideas and referenced direct quotations from their work (in the form of books, articles, essays, dissertations, and on the internet). No material other than that listed has been used.

Tübingen, August 1, 2016

Björn Rudzewitz

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Philosophische Fakultät

Name:

Vorname:

Matrikel-Nummer:

Adresse:

Hiermit versichere ich, die Arbeit mit dem Titel:

____

im Rahmen der Lehrveranstaltung ____

im Sommer-/Wintersemester ________________ bei ___________________________________

selbständig und nur mit den in der Arbeit angegebenen Hilfsmitteln verfasst zu haben.

Mir ist bekannt, dass ich alle schriftlichen Arbeiten, die ich im Verlauf meines Studiums als Studien- oder Prüfungsleistung einreiche, selbständig verfassen muss. Zitate sowie der Gebrauch von fremden Quellen und Hilfsmitteln müssen nach den Regeln wissenschaftlicher Dokumentation von mir eindeutig gekennzeichnet werden. Ich darf fremde Texte oder Textpassagen (auch aus dem Internet) nicht als meine eigenen ausgeben.

Ein Verstoß gegen diese Grundregeln wissenschaftlichen Arbeitens gilt als Täuschungs- bzw. Betrugsversuch und zieht entsprechende Konsequenzen nach sich. In jedem Fall wird die Leistung mit **„nicht ausreichend“ (5,0)** bewertet. In schwerwiegenden Fällen kann der Prüfungsausschuss den Kandidaten/die Kandidatin von der Erbringung weiterer Prüfungsleistungen ausschließen; vgl. hierzu die Prüfungsordnungen für die Bachelor-, Master-, Lehramts- bzw. Magisterstudiengänge.

Datum: _________________ Unterschrift: __________________________________

Contents

Abstract

Question answering is the task of automatically finding an answer to a question posed in natural language. In the digital age its potential can not be dismissed: with an ever-increasing amount of information available on the world wide web, it becomes more and more important for computers to be able to answer questions posed by human non-expert users, since it is not possible for humans to screen all information theoretically available to them.

The field of automatic question answering has diversified itself due to different scenarios in which question answering plays a role. For the present study two domains of question answering are compared in order to explore which information sources are general to question answering, and which ones are a product of domain adaptation. On the one hand this thesis analyzes which features are most effective for traditional question answering. In this case a system distinguishes between relevant and irrelevant answers to a question where the answers are provided in isolation from each other and the language is relatively close to standard language. On the other hand community question answering is represents a variant of question answering where the answers to a question are given in a thread structure and crawled from web forums, resulting in a range of web-specific artifacts.

For this thesis standard evaluation resources from each domain are given as input to a question answering system built to extract over 250 different features from five information sources: question features, answer features, question-answer features, answer-answer features, and user features. The feature sets are given to a logistic regression classifier in order to evaluate which features are most effective for which task.

The results of the experimental testing in a comparative setup show that for traditional question answering features that model the relation between a question and an answer are most effective, whereas for the case of community question answering systems benefit more from analyzing the properties of a specific answer and its role in the conversational context. The study experimentally confirms that domain adaptation is highly effective for community question answering and that traditional question answering and community question answering, despite looking similar on the surface, pose very different needs to question answering systems.

List of Figures

List of Tables

List of Abbreviations

e.g.	exempli gratia (for example)
et al.	et alii/aliae (and others)
cf.	confer
i.e	id est (that is/this means)
URL	Uniform Resource Locator (web address)

1 Introduction

Asking questions and searching for answers is a supremely human behavior. The process of searching for answers to questions includes cogitation and the consultation of different information sources. One of the biggest information sources nowadays is the world wide web. The information is however of limited use if it is not possible to pose queries (i.e. ask questions) in order to obtain data that is potentially relevant for answering a given question. Once a answer candidate selection has been performed it has to be decided which answer candidate is relevant given this question. A computer that has the task of answering a question thus has to have an understanding of what a question asks for, what characteristics answers have to fulfill, and how the relation between the question and a potentially relevant answer can be characterized.

The field of automatic question answering is concerned with the challenge of how a computer can retrieve an answer to a question given in natural language by a human. This is a wide-ranging task that spans over a range sub tasks that, due to their complexity, evolved into sub fields that continue to attract much research on their own (Peng, Lee, & Ingersoll, 2002). As described by Punyakanok, Roth, and Yih (2004) the basic sub tasks include question analysis, candidate document retrieval, and answer selection.

In the first step the question needs to be analyzed in order to determine what information the question asks for and how it asks for it. This is a necessary step for the second stage, the candidate retrieval. In this step a system retrieves documents that are potentially relevant for answering the question. These candidates are used in the last step in which these candidates are evaluated further. There are different scenarios such as candidate ranking, distinguishing relevant from irrelevant answers, or generating an answer in natural language based on the found information.

Community question answering is an extension of question answering that pursues the goal to distinguish between relevant and irrelevant answers in a community-based context. Usually web forum threads are the subject of analysis (Màrquez et al., 2015; Nakov et al., 2016). In this setting, the set of candidate answer is already given and the task is to select only relevant answers that were given to this question, and omit the irrelevant answers. Like for question answering, the output can be given in different textual forms. Community question answering often poses additional difficulties stemming from the informal language used in many web forums from which the data is crawled.

From a general perspective this processing difficulty due to different language registers amounts to a domain adaptation problem. Often web forums are restricted to a certain topic, for example programming, but this topic can span over a wide range of sub domains with different vocabulary, in our example object-oriented programming, functional programming, different programming languages, just to name a few. This and the fact that each user has a different writing style opens up a wide space of possible inputs and demands for systems that can handle previously unseen data on all linguistic levels of analysis.

The goal of this thesis is to explore the relationship between question answering and community question answering. More concretely the study aims at determining which information sources are of general value for question answering and its sub domains, and which information sources are a fruitful artifacts of domain adaptation.

To this end, two standard shared evaluation resources were selected to form the basis for the information sources and adaptation experiments. To make the results comparable, all experimental factors were set in the most similar way for both data sets. Concretely, the same evaluation metrics, the same pre-processing, the same feature generation, and the same machine learning algorithms were used.

This thesis is structured as follows: in Section 2, the data is described in detail. Section 3 shows the technical implementation of the question answering system. In Section 4 the study shows both results and an interpretation thereof. The results are put into a wider context of related work in Section 5. Finally Section 6 discusses the main findings and draws conclusions from the main findings of this study.

2 Data

For the experiments two different data sets representative for the respective domain of question answering and community question answering were selected. On the one hand, the CQA-QL corpus (Màrquez et al., 2015) from the SemEval 2015 Task 3 shared task was used. This data set was designed for the task of community question answering, where the task is to distinguish between relevant and irrelevant answers in a sequence of answers given to a question. As a contrastive data set, the TREC data subset compiled by M. Wang, Smith, and Mitamura (2007) was given to the system. The TREC data set is different in that it doesn't represent a community question answering corpus, but rather a resource where answers to a question are provided in isolation.

Table 2.1 shows the number of question and answers for each subset of the data. Each data set comes in the form development set, training set, and test set. As can be seen in Table 2.1, the CQA-QL corpus provides a much larger amount of data than the TREC corpus. Despite the larger amount of data,

	CQA-QL			TREC		
	dev	train	test	dev	train	test
Number of questions	300	2600	329	65	56	68
Number of answers	1645	16541	1976	1117	180	1442

Table 2.1: Statistics about the data used in this work.

there are on average more answers to one specific question in the TREC data set (∅ 14.5) than in the CQA-QL corpus (∅ 6.4). In the following, each of the corpora will be described in detail, before the differences and similarities will be discussed in more depth.

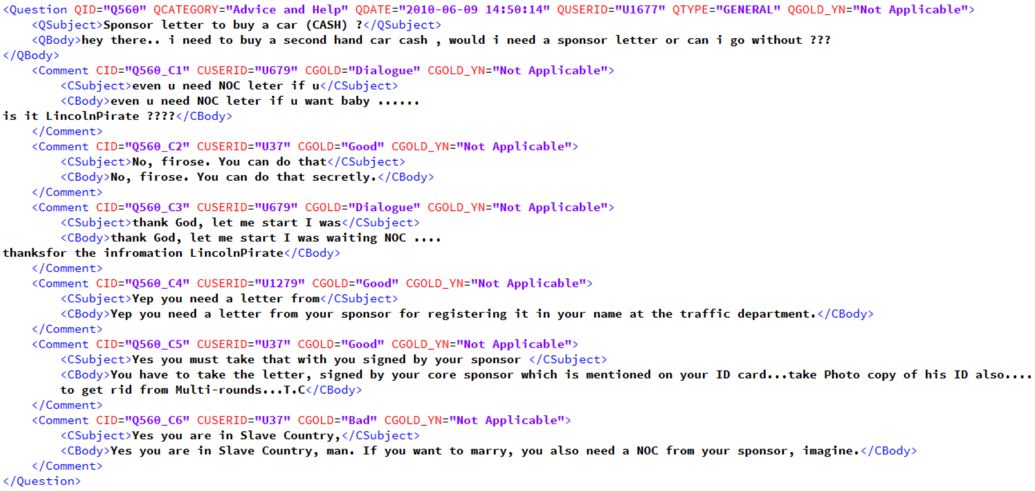
2.1 CQA-QL Corpus

The CQA-QL corpus (Màrquez et al., 2015) was first released to the public in the context of the shared task of community question answering at the 9th International Workshop on Semantic Evaluations (SemEval) 2015. Its basic structure consists of a two-level hierarchy: at the top level there are questions, and at the second level each question is associated with a list of one or more comments that were given as a response to this question by users.

The data was collected from the Qatar Living Forum[1], which provides a venue to ask and learn about the daily life in Qatar. From a computational linguistic perspective the data is both interesting and challenging, since not only does it contain a large amount of web-specific textual artifacts such as URLs, signatures, emoticons, e-mail addresses, spam, or typographical errors, but it also is a resource that contains a large amount of text written in English as a second language. This stems from the orientation of the forum to provide information to people from all over the world who want to move to or are new in Qatar.

Figure 2.1 shows an excerpt from the corpus. It shows various of the aforementioned challenging linguistic peculiarities of the data, which pose difficulties to automatic processing approaches. In the question there is an inconsistent mixture of lowercase and uppercase letters, e.g. the first person singular pronoun *I* is written in lowercase, whereas the word *CASH* (for cash) is written in all uppercase letters. As in the subsequent comments one can observe repeating adjacent punctuation symbols, e.g. the three question marks at the end of the question. In the first comment the second person singular pronoun *you* is expressed as an abbreviated form *u*. The token *letter*, spelled correctly in the question, can be found as an (incorrect) orthographic variant *leter* in the first comment. Other peculiarities are missing articles in the fifth comment, a signature at the ending of an answer (*T.C.*), and an address to another user (*LincolnPirate*).

[1] `http://www.qatarliving.com/forum` (last accessed 06/07/2016)



```xml
<Question QID="Q560" QCATEGORY="Advice and Help" QDATE="2010-06-09 14:50:14" QUSERID="U1677" QTYPE="GENERAL" QGOLD_YN="Not Applicable">
    <QSubject>Sponsor letter to buy a car (CASH) ?</QSubject>
    <QBody>hey there.. i need to buy a second hand car cash , would i need a sponsor letter or can i go without ???
</QBody>
    <Comment CID="Q560_C1" CUSERID="U679" CGOLD="Dialogue" CGOLD_YN="Not Applicable">
        <CSubject>even u need NOC leter if u</CSubject>
        <CBody>even u need NOC leter if u want baby ......
is it LincolnPirate ????</CBody>
    </Comment>
    <Comment CID="Q560_C2" CUSERID="U37" CGOLD="Good" CGOLD_YN="Not Applicable">
        <CSubject>No, firose. You can do that</CSubject>
        <CBody>No, firose. You can do that secretly.</CBody>
    </Comment>
    <Comment CID="Q560_C3" CUSERID="U679" CGOLD="Dialogue" CGOLD_YN="Not Applicable">
        <CSubject>thank God, let me start I was</CSubject>
        <CBody>thank God, let me start I was waiting NOC ....
thanksfor the infromation LincolnPirate</CBody>
    </Comment>
    <Comment CID="Q560_C4" CUSERID="U1279" CGOLD="Good" CGOLD_YN="Not Applicable">
        <CSubject>Yep you need a letter from</CSubject>
        <CBody>Yep you need a letter from your sponsor for registering it in your name at the traffic department.</CBody>
    </Comment>
    <Comment CID="Q560_C5" CUSERID="U37" CGOLD="Good" CGOLD_YN="Not Applicable">
        <CSubject>Yes you must take that with you signed by your sponsor </CSubject>
        <CBody>You have to take the letter, signed by your core sponsor which is mentioned on your ID card...take Photo copy of his ID also....
        to get rid from Multi-rounds...T.C</CBody>
    </Comment>
    <Comment CID="Q560_C6" CUSERID="U37" CGOLD="Bad" CGOLD_YN="Not Applicable">
        <CSubject>Yes you are in Slave Country,</CSubject>
        <CBody>Yes you are in Slave Country, man. If you want to marry, you also need a NOC from your sponsor, imagine.</CBody>
    </Comment>
</Question>
```

Figure 2.1: Example data from the CQA-QL corpus.

Each question and answer is associated with meta data. Table 2.2 based on (Màrquez et al., 2015) shows all fields provided for these elements. While features such as the question or answer identifiers serve as a structuring device of the corpus, other fields such as the question topic can provide useful information both for human users and machine learning approaches. Each of the (pre-defined) different topics is centered around the immigration to and life in Qatar and has to be selected by question inquirers before posting their questions. Examples are *Socialising*, *Working in Qatar*, or *Environment*. This topical variety underlines the need for robust systems that generalize across a range of topics.

Attribute	**Description**
Question	
QID	question identifier
QCATEGORY	one of 21 question categories defined by the forum
QDATE	question posting time stamp
QUSERID	user id of question inquirer
QTYPE	either GENERAL or YES/NO
QGOLD_YN	majority voting-based overall yes or no tendency for yes/no questions
Answer	
CID	answer identifier
CUSERID	user id of comment author
CGOLD	gold label of the comment
CGOLD_YN	yes/no gold labels for answers to a yes/no question

Table 2.2: Meta data attributes associated with each question and answer in the CQA-QL data set.

The presence of two different gold labels for each answer (CGOLD and CGOLD_YN) can be explained by the corpus design to support two parallel sub tasks of community question answering: answer selection and binary question answering. For the first task one of the values *Good*, *Bad*, *Potential*, *Dialogue*, *Not English*, or *Other* of the *CGOLD* attribute has to be predicted for each answer. For the task of binary question answering, systems not only have to predict one of the values *Yes*, *No*, or *Unsure* of the attribute *CGOLD_ YN*, but they also need to predict the QGOLD label of each question that indicates whether the yes/no question can be answered globally with *Yes*, *No*, or *Unsure*.

As Màrquez et al. (2015) describe the gold labels were obtained via crowd sourcing on Amazon Mechanical Turk. However, no details about any annotation guidelines are provided, and in the 2016 follow-up shared task on question answering (Nakov et al., 2016), the task of answer selection was modified in the sense that systems participating in the answer selection task only needed to distinguish between *Good* and *Bad* answers, with the original *Bad*, *Dialogue*, *Potential*, *Not English*, and *Other* labels all merged into one category *Bad*. This step was conducted due to inconsistencies in the original gold label annotation. This poses a significant problem to approaches aiming to predict the fine-grained answer labels. Due to this reason the present study reports results both for the binary and the multi-class gold labels.

2.2 TREC Corpus

The TREC question answering data set described compiled by M. Wang et al. (2007) is a collection of factoid questions together with corresponding answer candidates. This specific corpus was created on the basis of data provided for previous shared tasks of the Text Retrieval Conference (TREC). M. Wang et al. (2007) included only a subset of all answer candidates present in the original TREC data where at least one content word in the answer also occurs on the question.

Another difference to the original TREC data is that M. Wang et al. (2007) manually added human gold labels for each of the answers. However the article doesn't mention any annotation details such as the number, background, or agreement of the annotator(s).

Each answer is associated with a rank which indicates how well this answer answers the corresponding question. In addition, each answer bears a binary label that indicates whether this answer answers the question or not. The rank features can be explained by the corpus design to not only enable the task of answer selection, but also the task of acquiring and ranking relevant documents. The answers in the corpus don't stand in direct connection to each other, since they have been extracted from different documents.

Figure 2.2 shows an example with raw data from the corpus. The answers are ranked from most relevant to most irrelevant in decreasing order. The example shows that apart from numbers having been replaced by *<num>*, the language is very close to standard English.

```
Question:
what country is the biggest producer of tungsten ?
Answer 1:
china dominates world tungsten production and has frequently been
accused of dumping tungsten on western markets.
Answer 2:
then china, now the world's biggest producer, continued to pump
large quantities of the metal into export markets.
Answer 3:
even in china, the world's biggest producer, mine production has
more than halved since the late 1980s to less than <num> tonnes
in <num>.
```

Figure 2.2: Example from the TREC corpus.

2.3 Comparison of the Data Sets

Despite the fact that both the CQA-QL and the TREC data sets are question answering resources, they are very different in their design and nature. The CQA-QL was compiled as a realistic community question answering data set, whereas the purpose of the TREC corpus is to serve as a resource for answer acquisition and re-ranking with standard language.

One main difference arising from the data sources is the linguistic characterization of the language in the data sets. While the language in the CQA-QL data set exhibits very strong deviations from standard English and even posts in a language other than English, the language in the TREC corpus is closer to standard English. The CQA-QL corpus contains web-specific phenomena such as URLS, e-mails, emoticons, etc. which are not present in the TREC corpus.

The answers in the CQA-QL corpus are also mostly produced by non-native speakers of English, whereas the language in the TREC documents hints at either native speakers or non-native speakers with a high proficiency levels. Therefore, the CQA-QL data shows transfer effects where non-native speakers of English transfer linguistic knowledge from their first language into their second language (English), resulting in ungrammatical sentences, non-existing words, or uncommon word sequences.

Although both corpora are open-domain question answering corpora, the topic distribution differs. While the questions in the CQA-QL corpus are centered around the daily life in and immigration to Qatar and often ask for personal experiences or recommendations, the questions in the TREC data set are of a more factoid nature and don't include inquiries for personal responses. The more personal nature of answers in the CQA-QL corpus is also promoted by the thread structure of the answers: while in the CQA-Ql corpus users can be observed to react to previous posting by other users, the answers in the TREC data set stem from different documents and don't contain references to other answers.

The corpora have in common that their label distribution is highly imbalanced. In the CQA-QL corpus more than half of the comments are labeled as *Good*. The other class with many instances is the class *Bad*, which forms about 40% of the labels. The other 10% of the instances have one of the remaining four labels, with *Other* occurring exactly 3 times in over 20,000 instances. In the TREC data, about one quarter of the answers are correct, and the remaining answers are incorrect. This class imbalance poses difficulties to many machine learning algorithms and therefore has to be considered as a factor influencing the experimental results.

The CQA-QL corpus contains a significantly bigger amount of data than the TREC data. While this provides machine learning systems with more instances to learn from, it also raises the problem of data sparsity, since naturally the feature space will contain many zero values for big amounts of highly variable data.

The two data sets were chosen for the experiments in this study since they represent freely available standard evaluation resources and are from different domains and therefore allow for domain adaptation and domain transfer experiments, as well as insights about the in-domain effectiveness of systems.

3 Technical Implementation

The question answering system was implemented as a pipeline that incrementally enriches input data with annotations on multiple levels. These annotations are used for the extraction of a wide range of features which are given to a machine learning system that learns to distinguish between relevant and irrelevant answers. Figure 3.1 shows the structure of the main components of the system, which will be explained in detail in the following subsections.

At the top level there are two yellow nodes which represent the input given to the system. The input comes in the form of raw text, i.e. there exists no linguistic markup at this step. Dependent on the corpus, this raw text can contain meta data, such as user or question identifiers. Furthermore the data contains information about the classes which have to be predicted by the system, henceforth referred to as gold labels.

3.1 Precomputation Module

The first component in the pipeline is a Python module that allows to extract and precompute (the basis for) non-continuous features. Non-continuous

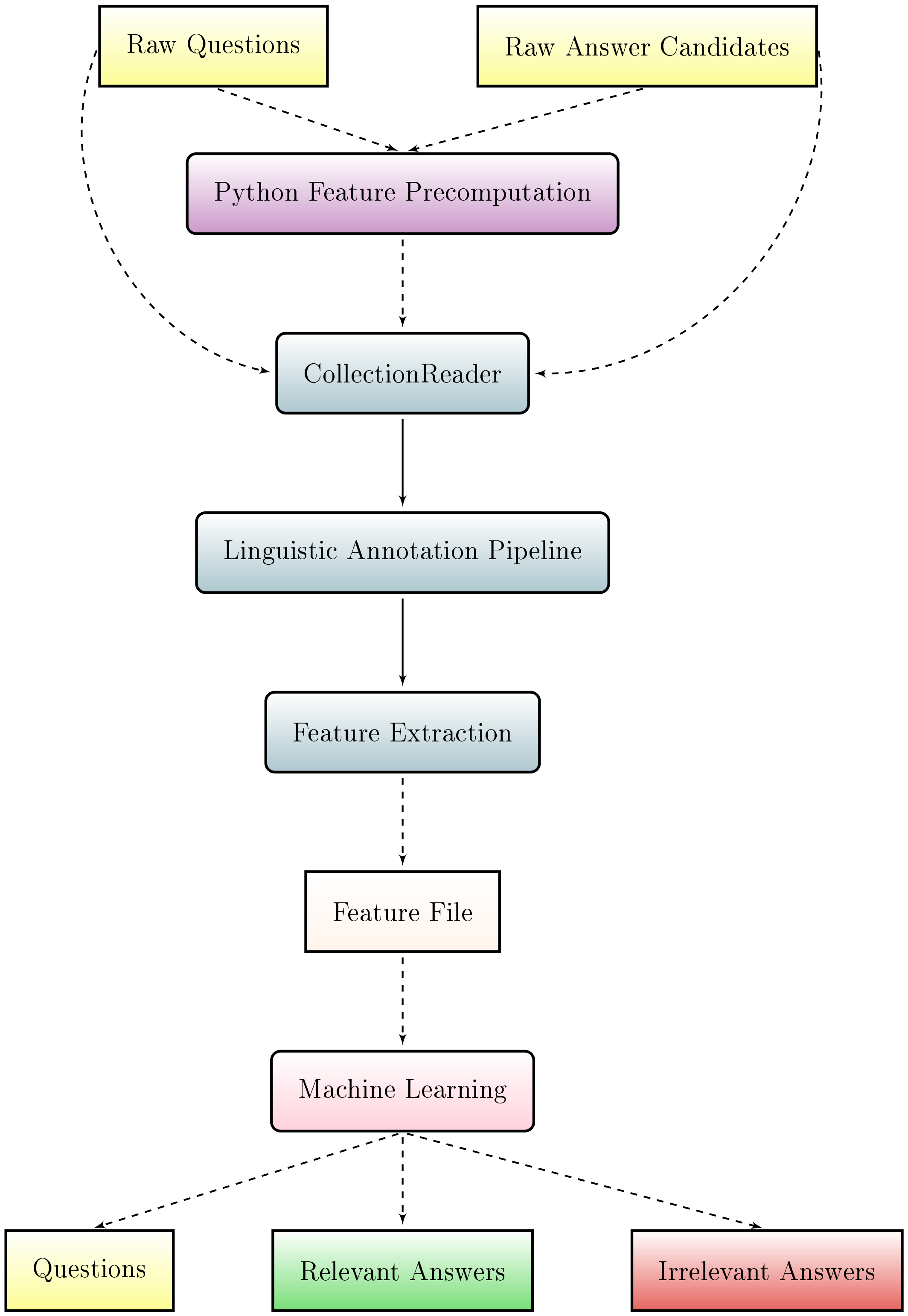


Figure 3.1: Structure of the main question answering pipeline.

features are features that encode information that doesn't directly express a characteristic of the question, answer, or question-answer pair. Examples are a feature that indicates how many relevant answers a specific user has provided in the whole training data, or a feature that indicates the relative position of an answer in a thread for a given question. This extra module is necessary because due to the architectural design of the main pipeline, the system only has access to either one question-answer pair or one answer-answer pair at runtime. However, by precomputing the relevant statistics for non-continuous features, these files can be read by the main pipeline, which enables the main component with an artificial look-ahead and look-back over the data.

3.2 Main Annotation Pipeline

The main pipeline is then the next step in the question answering system. All components belonging to the main system are shaded in blue in Figure 3.1. Not only does the main system take the feature precomputations from the previous module as input, but it also reads the raw questions and answers into the system. The main system is a UIMA (Ferrucci & Lally, 2004) pipeline. In order to understand the subsequent steps, it is necessary to briefly explain the main concepts of UIMA in the following.

The Unstructured Information Management Architecture (UIMA, (Ferrucci & Lally, 2004)) is a framework for large-scale, multi-layer stand-off annotation and processing of data. Its purpose is to provide a a means to assign structure to unstructured information such as text. It is however not restricted to annotating text, but also suitable for the annotation of unstructured information of other modalities such as images, videos, and more. The core idea of UIMA is to provide one common annotation index (*Common Analysis Structure*, *CAS*) for every object of analysis (*view*), which is passed through a series of modules that incrementally enrich this representation with module-specific information. In the complete process the original input is not altered, but information is only always added in the form of multiple layers of independent or dependent annotations. This property makes the framework especially suitable for natural language processing because in the processing pipeline every level of analysis is always reconstructable and accessible.

The UIMA module responsible for collecting data is the *CollectionReader*. It reads both the raw questions and answers as well as the additional information generated in the Python module. The CollectionReader generates an index over question-answer pairs and passes always one question-answer pair to the pipeline, for each answer. In technical terms it generates on CAS for the question-answer pair and two views: a question view, and an answer view. Subsequent components can request both access to on view, as well as to both views in parallel, dependent on the task. In the case where answer-answer features are extracted an additional view with the previous answer is added.

In the following a range of analysis modules are run that add information to the CAS. Tab 3.1 provides an overview about all annotation modules that operate on the CAS representation. The components are a mixture of components from the UIMA-based DKPro (Eckart de Castilho & Gurevych, 2014) toolkit, as well as native CoMiC components. The CoMiC system (Meurers, Ziai, Ott, & Kopp, 2011; Rudzewitz & Ziai, 2015) served as the technical starting point for the present study. The order of the components in the table from top to bottom reflects their usage in the UIMA pipeline. This order is determined by the specific needs of the tools, for example the dependency parser relies on sentence- and token-segmented input.

The first column of Table 3.1 shows each of the specific annotation tasks. Listed in the second column are the specific realizations of the annotation tasks. While the majority of the tools represent state-of-the-art natural language processing tools, the *Free Alignment* component deserves special attention. In this component, mappable annotations of different kinds are aligned between the question and answer. In the free alignment, in contrast to for example the Traditional Marriage Algorithm (Gale & Shapley, 1962) which was traditionally employed in the CoMiC system (Meurers, Ziai, Ott, & Kopp, 2011; Meurers, Ziai, Ott, & Bailey, 2011; Rudzewitz & Ziai, 2015), the free alignment approach doesn't use a givenness filter that excludes material from the alignment. It also doesn't impose a hierarchy of alignment types, but rather represents a greedy matching on various levels of linguistic abstraction (token, lowercase token, lemma, synonyms, semantic types, spelling, chunk, dependency triples). The third column indicates whether the respective UIMA wrapper class for the annotation tool was taken from DKPro or implemented in CoMiC.

Another tool worth mentioning is the *Web Token Annotator*. This tool was developed as an additional tokenizer that adds one additional token layer for web-specific tokens. It detects and annotates emoticons, hash tags, @-mentions (user back references), and URLs. This component was included in the pipeline due to the web-specific challenges of the CQA-QL data set.

UIMA requires the declaration of an explicit *type system* with all possible annotations and their types of values before any computation. In order to have a flexible annotation framework that is at the same time tailored towards the specific task of question answering, for this work the complete DKPro type system was merged with the alignment-specific annotation types of the CoMiC system. The advantage of this approach over unitary type systems is that all DKPro components use a shared type system, which makes it trivial to switch between different implementations of an annotation tool, while the alignment

Task	Tool	Implementation
Sentence Detection	OpenNLP[2]	DKPro
Tokenization	OpenNLP	DKPro
Web Token Annotation	ArkTweetNLP (Gimpel et al., 2011)	CoMiC
Part of Speech Tagging	TreeTagger (Schmid, 2013)	DKPro
Lemmatization	TreeTagger (Schmid, 2013)	DKPro
Chunking	OpenNLP	DKPro
Dependency Parsing	MaltParser (Nivre et al., 2007)	CoMiC
Semantic Role Labeling	NLP4J[3]	DKPro
Named Entity Recognition	OpenNLP	DKPro
Synonym Annotation	WordNet (Miller, 1995)	CoMiC
Semantic Type Annotation	WordNet (Miller, 1995)	CoMiC
Spelling Correction	CoMiC	CoMiC
Token Alignment	Free Alignment	CoMiC
Chunk Alignment	Free Alignment	CoMiC
Dependency Alignment	Free Alignment	CoMiC

Table 3.1: Annotation tasks and technical realization.

functionality is always maintained and independent of the specific DKPro tool.

3.3 Feature Extraction

The system extracts 251 different features from six conceptually different classes: question features, answer features, user features, question-answer features, answer-answer features, and meta features.

From a technical perspective each feature is extracted by a specific Java feature extractor class object that implements one interface shared by all feature extractors. Each feature extraction class is instantiated only once and computes its respective value with a static method. Due to the unified architecture of all feature extraction classes, the respective extractors can be loaded dynamically from a string value. The only input parameters to the feature extraction method is thus the name of the output file and a list of strings that represents the features to be extracted in a specific order. The feature extraction module outputs the feature values directly in WEKA's ARFF format (Hall et al., 2009).

In the following subsections, the different feature families, as well as their specific members are presented.

3.3.1 Question Features

Table 3.2 shows all question features used by the system. The question features are computed only on the question itself, without any reference to the answers. The motivation for these features is that a question serves as the basis for answers, and therefore needs to be characterized by the system. The linguistic well-formedness and semantics of the question influence the information structure of the answers given to the question (Krifka, 2008). In the following the specific features will be discussed in the same order as presented in Table 3.2.

The first set of question features indicates the presence of named entities in the question. This is relevant because when a named entity occurs in the question, then the respective named entities are introduced into the discourse, and answers might refer to these entities later.

Of great importance are also the question type features which indicate the presence or absence of certain English question words. As for example Meurers, Ziai, Ott, and Kopp (2011) showed, the question type has a strong influence on the nature and accuracy of answers given to the question. Certain question types such as "why" or "how" questions open a wider input space and therefor enable higher variation in the answers than yes/no questions which restrict the input to a binary decision. For this work the system distinguishes between seven basic question types, as shown in the table.

The next set of features are stylometric features that serve as an approximation of the style of the question. The three sentence length features are traditional readability features (Vajjala & Meurers, 2012). The reasoning is that questions with longer sentences might be more difficult to read and understand, influencing the nature of the answers. The remaining stylometric question features are character-based features based on (Stamatatos, 2009). An over-proportional usage of characters from a certain character class (such as uppercase letters) is an indicator of the style of the question and of the author of the question (Stamatatos, 2009).

The system furthermore re-uses part of speech features from the question answering system by Rudzewitz and Ziai (2015). These features indicate the distribution of part of speech tags of aligned material. As described in section 3.2, the system computes a multi-level alignment between the question and answer. The part of speech features express the distribution of syntactic classes over the aligned elements. The motivation is that alignments between certain elements (for example nouns) are more important than alignments between other elements (for example determiners) for answering a question.

Since in a question answering not only the question but also answers are important to characterize automatically, all answer features will be shown in the next section.

3.3.2 Answer Features

The answer features are parallel to the question features in that they express a single-view perspective on the data, but in this case only the answer is considered and characterized automatically by a range of features.

The first set of answer features are named entity features. They encode the presence and type of named entities in the answer string. The detection of named entities is potentially valuable in combination with question type

Feature	Description
Named Entity Features	
QuestionTotalNumNEs	total frequency of named entities
QuestionNumDateNEs	frequency of “date” named entities
QuestionNumLocationNEs	frequency of “location” named entities
QuestionNumOrganizationNEs	frequency of “organization” named entities
QuestionNumPersonNEs	frequency of “person” named entities
QuestionNumTimeNEs	frequency of “time” named entities
Question Type features	
QuestionWhat	binary presence of “what”
QuestionWhen	binary presence of “when”
QuestionWhere	binary presence of “where”
QuestionWhich	binary presence of “which”
QuestionWho	binary presence of “who”
QuestionWhy	binary presence of “why”
QuestionHow	binary presence of “how”
Stylometric Sentence Features	
QuestionMinSentenceLen	minimal question sentence length
QuestionMaxSentenceLen	maximal question sentence length
QuestionAvgSentenceLen	average question sentence length
Stylometric Character Features	
QuestionPropUppercase	proportion of uppercase letters in question
QuestionPropLowercase	proportion of lowercase letters in question
QuestionPropNumbers	proportion of digits in question
QuestionPropPunctuation	proportion of punctuation in question
QuestionPropNonASCII	proportion of non-ASCII letters in question
QuestionNumUppercaseWords	proportion of words starting with uppercase letters in question
QuestionContainsQuestionMark	binary presence of question mark in question
QuestionContainsExclamationMark	binary presence of exclamation mark in question
QuestionLongestAdjacentSequence	longest adjacent character sequence in question
QuestionTTR	type-token ratio of question
Web Text Features	
QuestionNumEmoticons	number of emoticons in question
QuestionNumHashtags	number of hash tags in question
QuestionNumAts	number of @-mentions in question
QuestionNumUrls	number of URLs in question
Part of Speech Features	
QuestionPropConjunction	proportion of words tagged as conjunction in question
QuestionPropCardNumber	proportion of words tagged as cardinal numbers in question
QuestionPropDeterminer	proportion of words tagged as determiner in question
QuestionPropExThere	proportion of words tagged as existential there in question
QuestionPropForeignWord	proportion of words tagged as foreign words in question
QuestionPropPreposition	proportion of words tagged as preposition in question
QuestionPropAdjective	proportion of words tagged as adjective in question
QuestionPropListMarker	proportion of words tagged as list markers in question
QuestionPropModal	proportion of words tagged as modals in question
QuestionPropNoun	proportion of words tagged as nouns in question
QuestionPropPredeterminer	proportion of words tagged as pre-determiners in question
QuestionPropPossesive	proportion of words tagged as possessives in question
QuestionPropPronoun	proportion of words tagged as pronouns in question
QuestionPropAdverb	proportion of words tagged as adverbs in question
QuestionPropParticle	proportion of words tagged as particle in question
QuestionPropEndPunctuation	proportion of words tagged as punctuation in question
QuestionPropSymbol	proportion of words tagged as symbols in question
QuestionPropInterjection	proportion of words tagged as interjection in question
QuestionPropVerb	proportion of words tagged as verb in question
QuestionPropWhDeterminer	proportion of words tagged as wh-determiner in question
QuestionPropWhPronoun	proportion of words tagged as wh-pronoun in question
QuestionPropPossessivePronoun	proportion of words tagged as possessive pronoun in question
QuestionPropWhAdverb	proportion of words tagged as wh-adverb in question
QuestionPropJoiner	proportion of words tagged as joiner in question
QuestionPropCurrency	proportion of words tagged as currency in question
Chunk Tag Features	
QuestionChunkProportionADJP	proportion of chunks tagged as ADJP in question
QuestionChunkProportionADVP	proportion of chunks tagged as ADVP in question
QuestionChunkProportionINTJ	proportion of chunks tagged as INTJ in question
QuestionChunkProportionNP	proportion of chunks tagged as NP in question
QuestionChunkProportionPP	proportion of chunks tagged as PP in question
QuestionChunkProportionPRT	proportion of chunks tagged as PRT in question
QuestionChunkProportionSBAR	proportion of chunks tagged as SBAR in question
QuestionChunkProportionVP	proportion of chunks tagged as VP in question

Table 3.2: Question features

features since certain question types may require a named entity of a certain type in the answer. For example a “who” question is likely to expect a named entity of type person in an answer. The presence or absence of certain named entities in the answer could thus be indicative of whether the answer addresses the question or not. However, this relation can not be encoded directly in the answer feature set since the answer is treated in quasi-isolation here.

The term quasi-isolation is used here because the next feature in the table indicates the relative position of the answer in the thread. This feature can only be extracted for the CQA-QL data set because of the lack of a thread structure in the TREC data. This feature is hypothesized to be useful for questions with a long list of answers. In the CQA-QL corpus, the maximal number of answers per question is 143, allowing the possibility of sub-threads evolving around other topics inside one thread.

Parallel to the question feature set, the answer feature set contains stylometric sentence length and character features. They serve as coarse predictors of the answer style and readability.

Taken from Rudzewitz and Ziai (2015) are the web text features. Emoticons are expressions of the sentiment of the answer author. Since the goal of question answering is to solve questions, a negative sentiment might arise from the expression of the inability to answer a question. In contrast, the interpretation of the presence of URLs is less straightforward. On the one hand, URLs can be useful links to external information which can help answering the question. For example a “where” question could ask for an information source, which could be satisfied with the provision of an URL. On the other hand, URLs are also a common means for spreading spam. In an open forum such as the CQA-QL corpus’ basis forum, users can easily post spam messages with links. This led for example Vo, Magnolini, and Popescu (2015) to the decision to employ a spam classifier on the CQA-QL data set. The web-specific hash tag and @-mention features are predictors of dialogue and topic, since hash tags are used to emphasize a certain topic, and @-mentions are explicit back references to previous users.

In addition to the web-specific features, the system implements a range of cohesion marker features[4] from different categories. These features can be indicative of the status of the thread answers. For example cohesion markers from the “Alternative” category such as “*alternatively*” or “*on the other hand*” are indicative of a contrasting view towards previous answers. Cohesion markers from the “summary” category such as “to conclude” or “in brief” can indicate that one or more answers to the question have been found, the content of which is summarized in the current answer.

[4] words and categories extracted from `http://library.bcu.ac.uk/learner/writingguides/1.33.htm` (last accessed 07/06/2016)

The last category of features in the first part of the answer features represents a set of features that indicate the presence of the most common words for each gold label, as observed in the training data. The intuition is that certain words are more common for irrelevant than relevant answers, at least for the community question answering data.

Feature	Description
Named Entity Features	
AnswerTotalNumNEs	total number of named entities in the answer
AnswerNumDateNEs	total number of "date" named entities in the answer
AnswerNumLocationNEs	total number of "location" named entities in the answer
AnswerNumOrganizationNEs	total number of "organization" named entities in the answer
AnswerNumPersonNEs	total number of "person" named entities in the answer
AnswerNumTimeNEs	total number of "time" named entities in the answer
Answer Position Features	
AnswerPositionInThread	relative position of answer in thread (CQA-QL only)
Stylometric Sentence Features	
AnswerMinSentenceLen	minimal sentence length of the answer
AnswerMaxSentenceLen	maximal sentence length of the answer
AnswerAvgSentenceLen	average sentence length of the answer
Stylometric Character Features	
AnswerPropUppercase	proportion of uppercase letters in answer
AnswerPropLowercase	proportion of lowercase letters in answer
AnswerPropNumbers	proportion of digits in answer
AnswerPropPunctuation	proportion of punctuation in answer
AnswerPropNonASCII	proportion of non-ASCII letters in answer
AnswerNumUppercaseWords	proportion of words starting with uppercase letters in answer
AnswerContainsAnswerMark	binary presence of answer mark in answer
AnswerContainsExclamationMark	binary presence of exclamation mark in answer
AnswerLongestAdjacentSequence	longest adjacent character sequence in answer
AnswerTTR	type-token ratio of answer
Web Text Features	
AnswerNumEmoticons	number of emoticons in answer
AnswerNumHashtags	number of hash tags in answer
AnswerNumAts	number of @-mentions in answer
AnswerNumUrls	number of URLs in answer
Cohesion Marker Features	
AnswerAlternativeCohesion	frequency of "alternative" cohesion markers in the answer
AnswerConcessionCohesion	frequency of "concession" cohesion markers in the answer
AnswerContrastCohesion	frequency of "contrast" cohesion markers in the answer
AnswerDeductionCohesion	frequency of "deduction" cohesion markers in the answer
AnswerExampleCohesion	frequency of "example" cohesion markers in the answer
AnswerGeneralizingCohesion	frequency of "generalizing" cohesion markers in the answer
AnswerHighlightingCohesion	frequency of "highlighting" cohesion markers in the answer
AnswerListingCohesion	frequency of "listing" cohesion markers in the answer
AnswerObviousCohesion	frequency of "obvious" cohesion markers in the answer
AnswerReformulationCohesion	frequency of "reformulation" cohesion markers in the answer
AnswerReinforcementCohesion	frequency of "reinforcement" cohesion markers in the answer
AnswerResultCohesion	frequency of "result" cohesion markers in the answer
AnswerSimilarityCohesion	frequency of "similarity" cohesion markers in the answer
AnswerSummaryCohesion	frequency of "summary" cohesion markers in the answer
AnswerTransitionCohesion	frequency of "transition" cohesion markers in the answer
Characteristic Words Features	
AnswerContainsTopBadWords	binary presence of most frequent words of "Bad" answers (CQA-QL only)
AnswerContainsTopDialogueWords	binary presence of most frequent words of "Dialogue" answers (CQA-QL only)
AnswerContainsTopGoodWords	binary presence of most frequent words of "Good" answers (CQA-QL only)
AnswerContainsTopNonEnglishWords	binary presence of most frequent words of "Not English" answers (CQA-QL only)
AnswerContainsTopOtherWords	binary presence of most frequent words of "Other" answers (CQA-QL only)
AnswerContainsTopPotentialWords	binary presence of most frequent words of "Potential" answers (CQA-QL only)
AnswerContainsAcknowledgment	binary presence of an acknowledgment marker
AnswerContainsSlang	binary presence of a slang marker

Table 3.3: Answer features (continued in table 3.4)

Due to space considerations, the second part of the answer features are presented in Table 3.4. These features are part of speech word and chunk features and express the distribution of part of speech tags over aligned elements. Since they are computed parallel to their question parts, please refer to Section 3.3.1 for a detailed explanation.

In the next section user features will be discussed, since every question and answer is written by a user who thus can be considered by the system.

Feature	**Description**
Part of Speech Features	
AnswerPropConjunction	proportion of words tagged as conjunction in answer
AnswerPropCardNumber	proportion of words tagged as cardinal numbers in answer
AnswerPropDeterminer	proportion of words tagged as determiner in answer
AnswerPropExThere	proportion of words tagged as existential there in answer
AnswerPropForeignWord	proportion of words tagged as foreign words in answer
AnswerPropPreposition	proportion of words tagged as preposition in answer
AnswerPropAdjective	proportion of words tagged as adjective in answer
AnswerPropListMarker	proportion of words tagged as list markers in answer
AnswerPropModal	proportion of words tagged as modals in answer
AnswerPropNoun	proportion of words tagged as nouns in answer
AnswerPropPredeterminer	proportion of words tagged as pre-determiners in answer
AnswerPropPossesive	proportion of words tagged as possessives in answer
AnswerPropPronoun	proportion of words tagged as pronouns in answer
AnswerPropAdverb	proportion of words tagged as adverbs in answer
AnswerPropParticle	proportion of words tagged as particle in answer
AnswerPropEndPunctuation	proportion of words tagged as punctuation in answer
AnswerPropSymbol	proportion of words tagged as symbols in answer
AnswerPropInterjection	proportion of words tagged as interjection in answer
AnswerPropVerb	proportion of words tagged as verb in answer
AnswerPropWhDeterminer	proportion of words tagged as wh-determiner in answer
AnswerPropWhPronoun	proportion of words tagged as wh-pronoun in answer
AnswerPropPossessivePronoun	proportion of words tagged as possessive pronoun in answer
AnswerPropWhAdverb	proportion of words tagged as wh-adverb in answer
AnswerPropJoiner	proportion of words tagged as joiner in answer
AnswerPropCurrency	proportion of words tagged as currency in answer
Chunk Tag Features	
AnswerChunkProportionADJP	proportion of chunks tagged as ADJP in answer
AnswerChunkProportionADVP	proportion of chunks tagged as ADVP in answer
AnswerChunkProportionINTJ	proportion of chunks tagged as INTJ in answer
AnswerChunkProportionNP	proportion of chunks tagged as NP in answer
AnswerChunkProportionPP	proportion of chunks tagged as PP in answer
AnswerChunkProportionPRT	proportion of chunks tagged as PRT in answer
AnswerChunkProportionSBAR	proportion of chunks tagged as SBAR in answer
AnswerChunkProportionVP	proportion of chunks tagged as VP in answer

Table 3.4: Answer features (continued)

3.3.3 User Features

Especially for the CQA-QL data set where user information is available the system encodes a range of user features as shown in Table 3.5. The first feature in the list is taken from (Hou et al., 2015) and is set to 1 if the author of the current answer is also the question author, otherwise it is set to 0. The other user features indicate how many answers, especially also of a certain type (i.e. with a certain gold label) a user has provided in the training data. Note that the training/development and test set are disjoint with respect to questions and answers, but not with respect to users. If they were not disjoint, these features would not be allowed since they would more or less directly encode the gold labels of the answers. However, given this setup, these features allow to encode the performance of a specific user in previously seen data. As for all forums there exist expert users or even moderators who give more than one relevant answer in multiple threads.

So far only one-directional features have been presented. In the next section, features will presented that encode the specific relations between a question and an answer, thereby taking into account two views on the data at the same time.

Feature	**Description**
Question Answer User Relational Features	
AnswerUserIsQuestionUserNumeric	binary indicator of equivalence of current answer and question user
AnswerUserFreqGoodAnswers	number of “Good” answers of this user in training data
AnswerUserFreqPotentialAnswers	number of “Potential” answers of this user in training data
AnswerUserFreqBadAnswers	number of “Bad” answers of this user in training data
AnswerUserFreqDialogueAnswers	number of “Dialogue” answers of this user in training data
AnswerUserFreqNotEnglishAnswers	number of “Not English” answers of this user in training data
AnswerUserFreqOtherAnswers	number of “Other” answers of this user in training data
AnswerUserFreqNonGoodAnswers	number of answers not tagged as “Good” of this user in training data
AnswerUserNumTotalAnswers	total number of answers of the current answer user in training data
AnswerUserNumTotalQuestions	total number of questions of the current answer user in training data
QuestionUserNumTotalQuestions	total number of answers of the current question user in training data
QuestionUserNumTotalAnswers	total number of questions of the current question user in training data
Positional User Features	
CurUserIsPrevUser	binary indicator of equivalence of current and previous answer user
CurUserIsNextUser	binary indicator of equivalence of current and next answer user
PrevUserIsQuestionUser	binary indicator of equivalence of current question and previous answer user
NextUserIsQuestionUser	binary indicator of equivalence of current question and next answer user

Table 3.5: User features

3.3.4 Question-Answer Features

The question-answer features encode a relation between a question and an answer. In contrast to all feature extractors discussed so far, the feature computation methods for the features listed in Table 3.6 use two UIMA views at the same time: question and answer. These features thus encode a relation of a specific question-answer pair.

The first set of features are n-gram similarity features. They express the cosine similarity of n-gram frequency vectors of the question and the answer. For each $n \in \{1, 2, 3, 4, 5\}$, the shared vocabulary of n-grams of this size is computed. For each vocabulary item, its frequency is entered into the respective vector. Once all frequencies for all vocabulary items have been computed, the system computes the cosine similarity between the two vectors, resulting in the feature value.

As Table 3.6 shows, suchlike n-gram similarity features are also computed on the lemma, part of speech, and character level. For chunks and dependency triples, the system computes frequency-based overlap features on the unigram level only due to the possible discontinuity of the features. For chunks there are two variants: in the first variant all chunks are considered, and in the

$$c_n(A,B) = \frac{|V(A,n) \cap V(Q,n)|}{|V(A,n)|}$$

Figure 3.2: N-gram containment measure.

second variant only content-bearing NP chunks are considered. This stems from the fact that the 'classical' CoMiC system only considers NP chunks.

The system additionally computes stylometric similarity features such as the cosine similarity between the answer and question type-token-ratio, as well as the average word length frequencies. The synonym, spelling, and interpolated features were taken from (Rudzewitz, 2016) after having proven to be highly effective for short answer assessment. The synonym and spelling overlap features express a percentage of greedily matched units on a merged level of surface forms and the respective abstract linguistic level conjoined via disjunction. The interpolated similarity measure is a textual similarity measure that computes the similarity of two texts in 49 dimensions of similarity compared via cosine similarity. These 49 dimensions span a space of lexical, character, syntactic, and semantic dimensions. For a detailed description, refer to Rudzewitz (2016).

Another measure computed by the system is the Longest Common Subsequence (LCS, (Myers, 1986)). It expresses the longest possibly discontinuous sequence shared by the question and the answer. In contrast, the greedy string tiling feature only considers the longest continuous sequence shared by the answer and the question.

The n-gram containment features are directional overlaps: as shown in for example (Clough & Stevenson, 2011), they are computed by dividing the size of the intersection of a shared vocabulary by the size of the vocabulary of one source document, in this case the answer. Figure 3.2 shows a formal definition of the n-gram containment measure. V stands for the vocabulary with $n \in \{1,2,3\}$ for this work. A is a specific answer, and Q a specific question. The formula was adapted from (Clough & Stevenson, 2011, page 13).

The Jaccard similarity measure used in the three Jaccard similarity features is similar to the n-gram measure. Instead of using a single vocabulary set in the denominator, it uses the intersection of both vocabularies as a normalization factor. For the n-gram containment, LCS, longest common substring, and Jaccard similarity features the corresponding DKPro method (Eckart de Castilho & Gurevych, 2014) is used in the system.

The Levenshtein distance feature expresses the number of insertion, deletion, or substitution operations needed to convert the answer to the question (Levenshtein, 1966), thereby representing an edit distance feature.

Finally Table 3.6 lists six features under the heading *Question Type Specific Constraint Fulfillment Features.* As the title suggests, these features encode certain question type specific constraints imposed on the answer. For example a *who* question requires a named entity of type *person* or *organization.* These features are especially useful for classifiers that don't explicitly model interactions between features.

Feature	**Description**
Word N-Gram Similarity Features	
WordUnigramOverlapGiven	cosine similarity of word unigram frequencies
WordBigramOverlapGiven	cosine similarity of word bigram frequencies
WordTrigramOverlapGiven	cosine similarity of word trigram frequencies
WordFourgramOverlapGiven	cosine similarity of word fourgram frequencies
WordFiveGramOverlapGiven	cosine similarity of word fivegram frequencies
Lemma N-Gram Similarity Features	
LemmaUnigramOverlap	cosine similarity of lemma unigram frequencies
LemmaBigramOverlap	cosine similarity of lemma bigram frequencies
LemmaTrigramOverlap	cosine similarity of lemma trigram frequencies
Chunk and Dependency Similarity Features	
ChunkOverlap	cosine similarity of chunk frequencies
NPChunkOverlap	cosine similarity of NP chunk frequencies
ChunkTagOverlap	cosine similarity of chunk tag frequencies
DepTripleOverlapRaw	cosine similarity of dependency triple frequencies
Part of Speech N-Gram Similarity Features	
POSUnigramOverlap	cosine similarity of part of speech unigram frequencies
POSBigramOverlap	cosine similarity of part of speech bigram frequencies
POSTrigramOverlap	cosine similarity of part of speech trigram frequencies
Character Similarity Features	
CharUnigramOverlap	cosine similarity of character unigram frequencies
CharUpperOverlap	cosine similarity of uppercase character unigram frequencies
CharLowerOverlap	cosine similarity of lowercase character unigram frequencies
CharBigramFreqOverlap	cosine similarity of character bigram frequencies
CharTrigramFreqOverlap	cosine similarity of character trigram frequencies
CharFourgramFreqOverlap	cosine similarity of character fourgram frequencies
CharFivegramOverlapFreq	cosine similarity of character fivegram frequencies
Various Similarity Features	
TTRSimilarity	cosine similarity of type-token-ratios
AvgWordLenghtOverlap	cosine similarity of average word length frequencies
SynonymOverlap	percentage of words with a synonym overlap
SpellingOverlap	percentage of words with a spelling overlap
InterpolatedOverlap	interpolated cosine similarity (Rudzewitz, 2016)
NamedEntityTagOverlap	percentage of overlapping named entity type
NamedEntityStringOverlap	percentage of surface named entities
DKPro Similarity Features	
UnigramContainment	word unigram containment
BigramContainment	word bigram containment
TrigramContainment	word trigram containment
UnigramJaccardSimilarity	word unigram Jaccard similarity
BigramJaccardSimilarity	word bigram Jaccard similarity
TrigramJaccardSimilarity	word trigram Jaccard similarity
GreedyStringTiling	longest sequence of greedy string tiling between question and answer
LevenstheinSimilarity	Levensthein distance from answer to question
LongestCommonSubsequence	longest common subsequence of question and answer
Question Type Specific Constraint Fulfillment Features	
QAWhatAndNE	binary presence of question word "what" and a named entity
QAWhenAndTimeNE	binary presence of question word "when" and a "date" named entity
QAWhereAndNE	binary presence of question word "where" and a "location" named entity
QAWhichAndNE	binary presence of question word "which" and a named entity
QAWhoAndNE	binary presence of question word "who" and a "person" or "organization" named entity
QAWhyAndNE	binary presence of question word "why" and a named entity

Table 3.6: Question-Answer features

3.3.5 Answer-Answer Features

Table 3.7 lists all answer-answer features. The feature names are the same as for the question answer features apart from the prefix *AnswAnsw*. This can be explained by the fact that the same feature value computation methods as described in Section 3.3.5 were used. The difference is that instead on using the question-answer pair as input, the input consists of the previous answer and the current answer. The features thus indicate the relation or similarity of the current answer to the previous answer.

For the first answer to each question, there exists no previous answer. In order to enable feature computations in this situation, the question is treated as the previous answer for the actual first answer. The answer-answer features can only be computed for the CQA-QL corpus due to the absence of a thread structure in the TREC corpus.

The motivation for using the answer-answer features is that a comparison with the previous answer can reveal topic changes and are hypothesized to be especially useful to detect dialogue and sub-threads under the main thread.

3.3.6 Meta Features

Table 3.8 shows four meta features used by the system. While the two identifier features are not used for predictions and are only included in case the feature files need to be augmented later, the two gold features are the actual values the system has to predict. There exist two variants because for the CQA-QL data set both a fine-grained gold label with six classes, as well as a binary diagnosis exist. For the machine learning experiments both labels have to be made explicit in order to be usable.

3.4 Machine Learning

For the experimental testing in this study logistic regression (cf. e.g. (Peng et al., 2002)) was employed. Logistic regression is a well-studied classification algorithm (Hosmer & Lemeshow, 2000). It lends itself especially well to binary classification tasks, since the output of the logistic regression is always a probability for choosing one out of two classes and it can cope well with class imbalance problems. For the cases with multiple outcome classes (the 6-way classification for the CQA-QL data set), the system uses a one-versus-rest approach. Since logistic regression works for binary classification, for each of the outcome classes a separate logistic regression classifier is trained that learns to distinguish between this class and all other remaining classes as a whole. For deriving a decision the prediction with the highest probability (i.e. the class most distant from the rest) is selected.

Feature	Description
Word N-Gram Similarity Features	
AnswAnswWordUnigramOverlapGiven	cosine similarity of word unigram frequencies
AnswAnswWordBigramOverlapGiven	cosine similarity of word bigram frequencies
AnswAnswWordTrigramOverlapGiven	cosine similarity of word trigram frequencies
AnswAnswWordFourgramOverlapGiven	cosine similarity of word fourgram frequencies
AnswAnswWordFiveGramOverlapGiven	cosine similarity of word fivegram frequencies
Lemma N-Gram Similarity Features	
AnswAnswLemmaUnigramOverlap	cosine similarity of lemma unigram frequencies
AnswAnswLemmaBigramOverlap	cosine similarity of lemma bigram frequencies
AnswAnswLemmaTrigramOverlap	cosine similarity of lemma trigram frequencies
Chunk and Dependency Similarity Features	
AnswAnswChunkOverlap	cosine similarity of chunk frequencies
AnswAnswNPChunkOverlap	cosine similarity of NP chunk frequencies
AnswAnswChunkTagOverlap	cosine similarity of chunk tag frequencies
AnswAnswDepTripleOverlapRaw	cosine similarity of dependency triple frequencies
Part of Speech N-Gram Similarity Features	
AnswAnswPOSUnigramOverlap	cosine similarity of part of speech unigram frequencies
AnswAnswPOSBigramOverlap	cosine similarity of part of speech bigram frequencies
AnswAnswPOSTrigramOverlap	cosine similarity of part of speech trigram frequencies
Character Similarity Features	
AnswAnswCharUnigramOverlap	cosine similarity of character unigram frequencies
AnswAnswCharUpperOverlap	cosine similarity of uppercase character unigram frequencies
AnswAnswCharLowerOverlap	cosine similarity of lowercase character unigram frequencies
AnswAnswCharBigramFreqOverlap	cosine similarity of character bigram frequencies
AnswAnswCharTrigramFreqOverlap	cosine similarity of character trigram frequencies
AnswAnswCharFourgramFreqOverlap	cosine similarity of character fourgram frequencies
AnswAnswCharFivegramOverlapFreq	cosine similarity of character fivegram frequencies
Various Similarity Features	
AnswAnswTTRSimilarity	cosine similarity of type-token-ratios
AnswAnswAvgWordLenghtOverlap	cosine similarity of average word length frequencies
AnswAnswSynonymOverlap	percentage of words with a synonym overlap
AnswAnswSpellingOverlap	percentage of words with a spelling overlap
AnswAnswInterpolatedOverlap	interpolated cosine similarity (Rudzewitz, 2016)
AnswAnswNamedEntityTagOverlap	percentage of overlapping named entity type
AnswAnswNamedEntityStringOverlap	percentage of surface named entities
DKPro Similarity Features	
AnswAnswUnigramContainment	word unigram containment
AnswAnswBigramContainment	word bigram containment
AnswAnswTrigramContainment	word trigram containment
AnswAnswUnigramJaccardSimilarity	word unigram Jaccard similarity
AnswAnswBigramJaccardSimilarity	word bigram Jaccard similarity
AnswAnswTrigramJaccardSimilarity	word trigram Jaccard similarity
AnswAnswGreedyStringTiling	longest sequence of greedy string tiling between previous answer and answer
AnswAnswLevenstheinSimilarity	Levensthein distance from answer to previous answer
AnswAnswLongestCommonSubsequence	longest common subsequence of previous answer and answer
Question Type Specific Constraint Fulfillment Features	
AnswAnswQAWhatAndNE	binary presence of previous answer word “what” and a named entity
AnswAnswQAWhenAndTimeNE	binary presence of previous answer word “when” and a “date” named entity
AnswAnswQAWhereAndNE	binary presence of previous answer word “where” and a “location” named entity
AnswAnswQAWhichAndNE	binary presence of previous answer word “which” and a named entity
AnswAnswQAWhoAndNE	binary presence of previous answer word “who” and a “person” or “organization” named entity
AnswAnswQAWhyAndNE	binary presence of previous answer word “why” and a named entity

Table 3.7: Answer-answer features

Feature	Description
Identifier Features	
QuestionId	identifier of the current question
AnswerId	identifier of the current answer
Gold Label Features	
GoldLabel	(potential multi-class) gold label
BinaryGoldLabel	binary gold label

Table 3.8: Meta features

Logistic regression is an extension of the linear regression model and thus often referred to as a generalized linear model (McCullagh & Nelder, 1989). It works by putting log odds into relation. An odd is computed by the dividing the probability of an outcome by the probability of not obtaining this outcome. An odds ratio is obtained by dividing the odds for two different conditions. A log odd (logit) is obtained by taking the logarithm of an odd (Peng et al., 2002). Borrowing the notation from Peng et al. (2002, page 2), the logistic regression model boils down to the following formula:

$$P(Y = y|X = x) = \frac{e^{\alpha+\beta x}}{1 + e^{\alpha+\beta x}}$$

Figure 3.3: Logistic regression model equation.

Y is the binary outcome variable and y a specific value of it. X with its specific instantiation x is the feature vector. For each dimension it contains the specific feature value for this specific training or testing instance. α and β are the model parameters (weights). While $\alpha \in \mathbb{R}$ is a single real-valued number, β constitutes a real-valued weight vector with the same dimensionality as the feature vector x. For each feature value the weight vector holds a specific weight which is multiplied with the feature value in the model and assigns a relative importance to this feature. The e terms give the functions its sigmoid shape, and the fraction represents an odd. The outcome is always between 0 and 1 and represents the probability P for obtaining the binary outcome value y given the model and the feature values.

4 Experimental Testing

This section reports the outcomes of the experimental testing. In order to understand the outcomes, the first subsection (4.1) proposes the evaluation metrics. The results shown in Section 4.2 are based on these metrics and interpreted in Section 4.3.

4.1 Evaluation Metrics

For the experimental testing, two evaluation procedures were chosen. On the one hand, the study reports accuracies for each data set when a classifier is trained on the combined training and development set and evaluated on a held-out test set. On the other hand a ranking of features by information gain is provided.

Accuracy is a metric used for evaluating predictions in a classification task. Figure 4.1 shows a formal definition of the accuracy metric (confer e.g. Bradley (1997)). It counts the number of true positives (TP) and true negatives (TN) and normalizes it by the total number of predictions.

$$\text{acc} = \frac{\#\text{TP} + \#\text{TN}}{\#\text{TP} + \#\text{TN} + \#\text{FP} + \#\text{FN}}$$

Figure 4.1: Formal definition of the accuracy metric.

Information gain is a metric that indicates how informative an attribute is with respect to deriving a certain categorical outcome. Figure 4.2 shows a formal definition of the information gain metric (Hall et al., 2009). It relates the entropy H of a class and the entropy of a class given an attribute.

Entropy, as depicted in Figure 4.3 from Renyi (1961), expresses the "amount of uncertainty concerning the outcome of an experiment" (Renyi, 1961, page 1). Given a distribution of a random variable with outcome probabilities $p_1, p_2, \cdots p_n$, the entropy metric represents a sum of the products of p_k with the logarithm of the inverse of p_k.

$$\text{info-gain(attribute, class)} = \text{H(class)} - \text{H(class} \mid \text{attribute)}$$

Figure 4.2: Formal definition of the information gain metric.

$$H(p_1, p_2, \cdots, p_n) = \sum_{k=1}^{n} p_k log_2 \frac{1}{p_k}$$

Figure 4.3: Formal definition of entropy as shown in (Renyi, 1961).

4.2 Evaluation Results

In the following the results of the evaluation are shown. In the tables the following abbreviations are used: *a* (answer), *q* (question), *qa* (question-answer), *u* (user), and *aa* (answer-answer). These abbreviations stand for the different feature sets presented in Section 3.3 and represent different experimental conditions under which the system is evaluated for a given data set.

Table 4.1 shows the system accuracy when trained on the training and development set of the CQA-QL corpus and tested on the CQA-QL test set.

The different conditions represented by the different rows are different feature sets, as explained in Section 3.3.

The results are reported for two different scenarios: for a binary classification (*good* vs. rest) following Nakov et al. (2016), as well as a 6-way classification as in Màrquez et al. (2015). The decision to report the outcome in two metrics was taken to make the results as comparable as possible to related work. For the TREC data set, the outcomes are reported for the binary classification task (*relevant* vs. *irrelevant* answer). The results are shown in Table 4.2.

In addition to reporting only numbers for the accuracy metric, Tables 4.4, 4.5, and 4.6 show a ranking of the 15 most informative features for each data set and classification task in terms of the information gain of each feature. The next section gives an interpretation of these results.

condition	**acc$_{\textbf{multi}}$**	**acc$_{\textbf{bin}}$**
random baseline	16.7	50.0
majority baseline	50.5	50.5
u	58.1	64.0
aa	51.0	57.5
qa	49.5	61.3
a	59.0	71.0
q	50.3	54.7
aa+u	58.6	64.3
qa+u	58.4	66.6
a+u	62.6	73.4
q+u	58.4	66.6
qa+aa	52.2	59.4
a+aa	59.7	71.3
q+aa	52.2	59.4
a+qa	59.6	70.5
q+qa	53.1	59.8
q+a	59.0	71.1
qa+aa+u	58.6	64.3
a+aa+u	59.7	71.3
q+aa+u	52.1	59.4
a+qa+u	59.6	70.5
q+qa+u	53.1	59.8
q+a+u	58.4	63.0
a+qa+aa	59.7	71.3
q+qa+aa	52.2	59.4
q+a+aa	52.17	59.4
q+a+qa	53.1	59.7
a+qa+aa+u	59.7	71.3
q+qa+aa+u	52.3	59.4
q+a+aa+u	52.1	59.4
q+a+qa+u	53.2	59.8
q+a+qa+aa	52.2	59.4
q+a+qa+aa+u	52.2	59.4

Table 4.1: Experimental results for different experimental conditions (feature groups) for the CQA-QL corpus. All values are reported in the accuracy metric in percentages.

Table 4.3 shows the accuracies obtained for cross-corpus and combined-corpus testing. In the column *TREC*, results are reported when the system is trained on the CQA-QL corpus and evaluated on the TREC data. The *CQA-QL* data shows results for a system when trained on the TREC data

condition	$\mathbf{acc_{bin}}$
random baseline	50.0
majority baseline	79.9
q	80.2
a	81.7
qa	82.5
q+a	82.5
q+qa	84.1
a+qa	82.5
a+q+qa	83.0

Table 4.2: Experimental results for different experimental conditions (feature groups) for the TREC data set. All values are reported in the accuracy metric in percentages.

and evaluated on the CQA-QL data. These columns thus cases thus represent cross-domain testing results. In the last column results are reported for the case where the system was trained on the combined TREC and CQA-QL data set and also tested on the combined test sets. In this case both the training and testing conditions are performed on mixed-domain data.

condition		testing on	
	TREC	CQA-QL	combined
random baseline	50.0	50.0	50.0
majority baseline	79.9	50.5	62.9
q	28.2	48.3	65.8
a	64.7	49.0	67.5
qa	73.4	50.5	65.2
q+a	55.5	49.5	74.9
q+qa	56.8	48.1	70.3
a+qa	72.7	50.5	70.5
a+q+qa	64.2	47.8	75.8

Table 4.3: Experimental results for different experimental conditions (feature groups) for cross-corpus and combined-corpus binary classification testing. All values are reported in the accuracy metric in percentages.

rank	feature	feature group
1	BigramJaccardSimilarity	qa
2	BigramContainment	qa
3	AnswerChunkProportionPP	a
4	UnigramJaccardSimilarity	qa
5	AnswerPropNoun	a
6	UnigramContainment	qa
7	GreedyStringTiling	qa
8	AnswerAvgSentenceLen	a
9	QuestionPropAdjective	q
10	InterpolatedOverlap	qa
11	AnswerPropPunctuation	a
12	QuestionTTR	q
13	QuestionPropParticle	q
14	AnswerMaxSentenceLen	a
15	AnswerPropPreposition	a

Table 4.4: Most informative features for the TREC data set.

rank	feature	feature group
1	AnswGreedyStringTiling	aa
2	AnswInterpolatedOverlap	aa
3	AnswLevenstheinSimilarity	aa
4	AnswUnigramJaccardSimilarity	aa
5	QuestionTTR	q
6	QuestionMaxSentenceLen	q
7	QuestionMinSentenceLen	q
8	AnswBigramJaccardSimilarity	aa
9	QuestionAvgSentenceLen	q
10	AnswBigramContainment	aa
11	AnswTTRSimilarity	aa
12	QuestionChunkProportionADVP	q
13	AnswLongestCommonSubsequence	aa
14	AnswUnigramContainment	aa
15	AnswPOSUnigramOverlap	aa

Table 4.5: Most informative features for the CQA-QL corpus for the binary classification task.

rank	feature	feature group
1	QuestionMaxSentenceLen	q
2	QuestionMinSentenceLen	q
3	AnswInterpolatedOverlap	aa
4	AnswGreedyStringTiling	aa
5	QuestionAvgSentenceLen	q
6	QuestionAvgSentenceLen	q
7	AnswLevenstheinSimilarity	aa
8	QuestionTTR	q
9	QuestionChunkProportionPP	q
10	AnswBigramJaccardSimilarity	aa
11	QuestionChunkProportionADVP	q
12	QuestionPropAdjective	q
13	AnswUnigramContainment	aa
14	AnswBigramContainment	aa
15	AnswTTRSimilarity	aa

Table 4.6: Most informative features for the CQA-QL corpus for the multi-class classification task.

4.3 Interpretation of Results

The results show different trends for the different data sets. For the TREC data set, features that characterize the relation between a question and an answer are most informative. This is reflected both in the accuracy table (Table 4.2) and the information gain ranking (Table 4.4). For this data set, question-answer features on their own yield the highest accuracy of feature groups when considered in isolation (82.5%). When considering all feature set combinations, the combined set of question and question-answer features perform best globally (84.1%).

The information gain ranking shows that the most informative features for this data set are n-gram overlaps of different sizes between the question and the answer (*BigramJaccardSimilarity*, *BigramContainment*, *UnigramJaccardSimilarity*). This and the fact that the *GreedyStringTiling* feature is ranked the seventh most informative feature shows that for the TREC data set it is useful and possible to search for longer string matches between a question and an answer. This implies that the language in the question and answer is comparable to a certain extent, since the most informative features are overlap features which operate on the surface representation. Among the non-question-answer features there are both features that model answer as well as question properties, as can be seen in the information gain ranking. Both for the question and answer characterization the system mostly relies on features that correlate with complexity and style, as can be seen in the features *AnswerChunkProportionPP*, *AnswerAvgSentenceLen*, *AnswerPropPunctuation*, and *QuestionTTR*.

The fact that the answer features only play a subordinate role in terms of accuracy when compared to the CQA-QL corpus can be explained by the absence of the extreme variability in answers that is prevalent in the CQA-QL corpus. Since the machine learner is not able to infer as much information from the answer style, it relies rather on other feature sets, as discussed above.

Noteworthy is also the *InterpolatedOverlap* measure by Rudzewitz (2016) on rank 10, a measure which is not only in the top 10 information gain rankings of all experiments in the present study, but also proved to be highly effective for the domains of short answer assessment and plagiarism detection. This shows that this similarity measure incorporating lexical, syntactic, semantic, and character features is highly generalizing and effective across different domains.

For the CQA-QL corpus the results show different trends. In contrast to the TREC data where question answer features are most effective and informative, the most informative features for the CQA-QL corpus clearly are answer features. While for all of the high accuracies above 70% for the binary classification task features modeling a single answer are included (see Table 4.1), the information gain ranking shows that for this data set and task especially features that relate the current answer to the previous answer are relevant. This means that by considering the characteristics of an answer, its role in the local discourse context, (and statistics about the user who wrote it), the classifier can learn valuable information.

The best result for both the binary classification task, as well as for the 6-class classification is obtained with a feature set combining answer and user features (62.2% and 73.4%, see Table 4.1). These accuracies are highly statistically significant when compared to the respective random (16.7% and 50.0%) and majority baseline (50.5%), with $p < 0.001$ (McNemar's test, cf. (Dietterich, 1998)).

This shows that domain adaptation is highly beneficial when moving from question answering to *community* question answering. Since the question answer features most effective for the TREC data set only play a minor role, and instead domain-specific user and discourse features are more effective, this comparison makes evident how crucial it is to consider the specific domain a system is applied to and to adapt the system correspondingly.

The information gain rankings in Table 4.5 and Table 4.6 show that even though certain differences depending on how many classes have to be distinguished can be observed for the CQA-QL data set, the overall trend for both data sets is that features expressing a relation between an answer and the previous answer are most informative, as is reflected in the various answer-answer similarity features prefixed with *Answ*. For both data sets the greedy string tiling and interpolated similarity measures are the most informative similarity measures. For the multi-class classification the two most informative features are stylometric features that indicates sentence length statistics of the question. This is an indicator that the question complexity has an influence on the nature of answers given as a response to it. One the one hand it is the case that questions with very long sentences are more difficult to process by humans, which can lead to imprecise answers. On the other hand answers with strong deviations from the standard language are also difficult to process by natural language processing tools. The tools might not be able to assign meaningful sentence boundaries in this case. The question length statistics can thus also be seen as a heuristics of the well-formedness of an answer, which again correlates with complexity and human processing difficulty.

Another observation that can be made when comparing the information gain ranking of the CQA-QL and TREC data sets is that for the CQA-QL data set bigram-based features are ranked lower than unigram-based features, thereby representing an inverse ordering. This raises the question of the comparability of the units under consideration. For the CQA-QL corpus it is more effective to compare atomic units (words) instead of sequences of units, such as bigrams. This and the fact that character-based edit distance features such as the *AnswLevenstheinSimilarity* feature on rank 3 (Table 4.5) and 7 (Table 4.6) are very informative confirm the observation that the language in the CQA-QL corpus is highly variable and distant to standard English.

For the mixed-domain testing the majority baseline drops when compared to the single-domain testing conditions since the combined data set is more balanced. For all conditions (feature sets) in the mixed-domain data set, the system outperforms the random and majority baseline. However, for the cross-domain testing the results are worse, with many accuracies below the random and majority baseline. This again confirms the observation that the community question answering and traditional question answering tasks are different and that out-of-domain testing poses difficulties to natural language processing systems.

5 Related Work and Background

Question answering has attracted a considerable amount of work over the last decade. In the following the key milestones in question answering will be summarized chronologically.

Punyakanok et al. (2004) proposed a dependency tree mapping between question and answer. With a tree edit distance they computed the cost for transforming one tree into another tree with the operations deletion, insertion, and substitution. Given a question and answer candidates, for each pair they computed the minimal tree edit distance and selected the answer with the minimal edit distance. Their proposal of the Approximate Tree Matching strategy laid the foundation for subsequent tree-based work on question answering. The idea is to match only a part of a tree and not increase the cost for deletion operations in order to account for question-answer pairs of different lengths. More concretely, after classifying the question type with a machine learning component they aimed at reformulating the question as a statement so as to facilitate the tree matching performed on the dependency level.

Several aspects from this work are relevant for the present work. While the matching in the present study is performed on various levels of linguistic abstraction including words, characters, lemmas, part of speech tags, word lengths, and dependency triples, a tree-based matching constitutes a promising next step and an information source worth exploring. Also the idea of classifying the type of a question is reflected in the present study. What is orthogonal to this work is the use of a machine learning component for deriving the question types. Further intrinsic evaluations are needed to determine the need of a machine learning component for question type classification.

Cui, Sun, Li, Kan, and Chua (2005) proposed fuzzy dependency matching instead of hard dependency matching, as done in the previously mentioned study. This step was implemented to allow for more flexibility which can arise from the non-obvious relations between the question and answers. The most important claim made by Cui et al. (2005) is that the direction of the dependency relations should be ignored since they often swap in questions and answers. This insight was used in the implementation of the system of the present study. Cui et al. (2005) reported high improvements in accuracy for the task of retrieving passages relevant for answering a question.

The work by M. Wang et al. (2007) is a seminal work for integrating machine translation insights into question answering. They proposed a quasi-synchronous grammar which models the relation between a question and an answer. This approach represents an early statistical, generative approach towards question answering. The data set used in this study became the standard evaluation resource for question answering for a decade and was also used in the present study.

Heilman and Smith (2010) extended the dependency tree representation for question answering by encoding dependency relations as complex objects consisting of the lemma, part of speech tag, and the dependency label of the corresponding word.

Following M. Wang et al. (2007), M. Wang and Manning (2010) showed a new method for computing an edit distance for question answering. In addition to the three commonly used edit operations insertion, deletion, and substitution they defined 43 specialized edit operations on different levels. Building on that they defined a feature set that models the dependency-based alignments. Examples are named entity alignments, typed dependency matches, fuzzy dependency matches, or semantic role overlaps. Some matching types like the named entity match are parallel to the matching types of the present study, partly because they had already been implemented in the CoMiC system (Meurers, Ziai, Ott, & Kopp, 2011), which formed the starting point of the present work.

Yao, Van Durme, Callison-Burch, and Clark (2013) modeled answers as chains of tokens where each token is assigned a tag indicating whether the token is relevant in this answer or not. This was achieved by a conditional random field sequence classifier. For the relation between the relevant part of an answer and the question Yao et al. (2013) defined a diverse set of features including semantic type matching, question types, part of speech n-grams, question types, and more features. Some of their features resemble the features employed in the present study.

Severyn and Moschitti (2013) showed that tree kernels are very useful for question answering. In the first step trees are created for a question and an answer. Then nodes with the same words are aligned. The goal is to align the question word and corresponding named entities in the answer. To determine which named entities are corresponding to the question word they built a classifier based on a taxonomy by Li and Roth (2002) to determine what the question asks for, such as locations or dates. Depending on whether a question-answer pair fulfills these constraints it is labeled as relevant or not. Their idea to encode question-type specific alignment constraints was used in the present study in the question-answer features, as explained in Section 3.3.

Chang and Pastusiak (2013) proposed a model that weights alignments between a question and answer. Words are aligned based on semantic relatedness as output by a WordNet (Miller, 1995) based approach and a distributional semantics model. They removed stop words from the question and answer prior to the alignment. The aligned words are then weighted by their inverse term frequency and the type of match (word/lemma/semantic type/named entity). This is directly comparable to the alignment features of the CoMiC system used in the present study.

Starting from 2014 neural networks have seen a strong rise in the domain of question answering. Yu, Hermann, Blunsom, and Pulman (2014) used a convolutional neural network that takes all words of a question and answer as input features, which are combined in a subsequent hidden convolution layer in order to find useful combinations of features (i.e. longer n-grams). In the work by D. Wang and Nyberg (2015) words are first converted into distributional semantics vectors, which are given as input to various types of different neural networks. They showed that this approach that doesn't require any pre-processing apart from the vector lookup could yield competitive results on the TREC data set compared to 'traditional' approaches.

Z. Wang, Mi, and Ittycheriah (2016) computed the similarity of a question-answer pair based on word embeddings. Individual embeddings are combined in order to derive a sentence-wide similarity representation to avoid the common word-based matching. While the neural network approaches have a strong potential for raising the upper bound for question answering, at this point they are not suitable for the present study because they are not fully interpretable, in contrast to the employed logistic regression. Since the goal of the present study is to gather qualitative insights into information sources and domain adaptation, neural network models were not employed here.

The work described here is directly related to the systems presented by participants of the 2015 and 2016 SemEval shared tasks on community question answering (Màrquez et al., 2015; Nakov et al., 2016). Due to the fact that the community question answering data set was released for the 2015 shared task and re-used for the 2016 follow-up shared task, both the systems and results are directly comparable to the present work. In the following, the approaches by participating systems will be discussed.

The main focus of the work by Yi, Wang, and Lan (2015) is to use answer characteristics for predicting whether an answer is useful for answering a question or not. One step towards achieving this goal is to detect non-English answers by comparing each word to a list of words generated from the various lexical resources in the NLTK distribution (Bird, 2006). In case the word doesn't occur in the NLTK list, a counter for this answer is increased. They implemented a heuristics that labeled each answer with more than 10 out-of-vocabulary items or a percentage of more than 60% unknown tokens as irrelevant for answering this question. This idea of using stylometric answer features as a source of information for question answering is parallel to the approach taken in the present work. Another parallel is the pre-processing: as in the present work Yi et al. (2015) saw a need for text normalization as a preprocessing step. While they, as in the present work, substituted HTML entities with the corresponding character, they also removed HTML tags, URLs, emoticons, signatures, repeating word repeating punctuation from the answers. While the removal of the web-specific elements can be seen as a step towards enabling basic NLP tools with a better basis for their analyses, this stands in contrast to the present work where use is made of these domain-specific answer properties in order to use them as a measure of the textual quality of the answer. Instead of using slang words directly as a feature, as done in the present work, Yi et al. (2015) detect and convert slang variants of words to their standard English counterparts. The authors extract a range of features that can be put into the category system applied for the present work: as previously mentioned, they employ a range of stylometric features for the answer classification, including the length of sentences, paragraphs, tokens, the length of the current answer answer divided by length of longest answer to this question, the presence of question marks, and the presence of "words of suggestion" (Yi et al., 2015). For the user features, they employ a range of features re-used in the present work: the number of correct answers of the answer author, and the number of correct answers of the answer author for this question category. The question-answer similarity features represent standard textual similarity features on the word, lemma, part of speech, character, and n-gram level. Worth mentioning are also the WordNet-based path similarity and the sentence-wide Latent Semantic Analysis. The authors furthermore encode both the given question category and the presence of question words, parallel to the present work. All these features are used in a Support Vector Machine setting, ad the authors found that combining all feature groups yielded the best results.

Vo et al. (2015) perform a similar slang normalization, but chose another similarity approach: similar to Heilman and Smith (2010), they decided to apply a tree similarity model to determine the similarity between a question and and answer. In addition, they encode alignments on the word level obtained by using the METEOR (Banerjee & Lavie, 2005) toolkit. One feature in their system can be seen as a user feature, namely a binary feature indicating the (non-)equivalence of the question and current answer author. In contrast to every other approach, they used a spam/ham classifier to detect answers which contain spam content. In their experiments they found out that their classifier ignores the minority classes for the multi-class classification task, probably caused by the strong class imbalance and questionable gold standard annotations.

Hou et al. (2015) used stylistic features of the answer such as the maximal word length, the average word length, the proportion of different word types (capital words, sentiment words, etc.), and character class proportions (including question marks, exclamation marks). Together with features expressing the proportions of part of speech tags, the number of named entities of different types, the frequency of URLs in the answer, and the relative position of the answer in the thread they intended to use as much information as possible of the answer itself for characterizing the quality of an answer given a question. The majority of the features proposed in this work were re-implemented in the present work. For the machine learning part, Hou et al. (2015) tested a hierarchical classification method against an ensemble learning setup, with better results for the hierarchical classifier.

The approach by Lin and Wang (2015) is interesting since they tested two powerful state-of-the-art machine learning algorithms for community question answering: neural networks, and conditional random fields. The usage of conditional random fields is worth mentioning since Lin and Wang (2015), among Rudzewitz and Ziai (2015), were the only participants of the 2015 shared task on community question answering that employed a sequence classification that makes use of the interdependence between answers and didn't treat them in isolation.

Tran, Tran, Vu, Nguyen, and Pham (2015) built the winning system of the SemEval 2015 task 3 challenge on community question answering with a system that obtained an accuracy of 73.76% for the multi-class classification. They trained a monolingual translation model with a Viterbi decoder that indicates to what degree an answer represents a 'translation' of the question. In addition they made use of a topic model with 100 different topics trained on the training data and a Wikipedia dump, with the variant trained on the training data proving more effective. The topical similarity was computed as the cosine similarity between the question and the answer. Tran et al. (2015) also applied the cosine similarity to compute a dependency, word, and noun phrase similarity between the question and the answer. Their idea to use feature encoding characteristic words for each outcome was re-used in the present work. Similar to other approaches, Tran et al. (2015) also computed alignment-based similarity features. For the machine learning, they used a relatively simple yet effective regression approach and converted the raw outcomes to labels.

From the work by Nicosia et al. (2015) the word n-gram similarity measures, as well as the Greedy String Tiling, Longest Common Subsequence, Jaccard Coefficient, and Word Containment measures were borrowed. Also the numerous meta data-related features indicating the relation between the question and answer author, as well as features expressing statistics about authors extracted from the training data are reflected in the present work. What differs though is the usage of tree kernels and support vector machines.

Meysam, Fakhrahmad, and Street (2015) were the only participants of the SemEval 2015 task 3 who decided to use decision trees for classifying answers. Interesting is also their decision to use boosting to balance the heavily imbalanced training data.

Just like in the present work, Belinkov, Mohtarami, Cyphers, and Glass (2015) made use of the DKPro (Eckart de Castilho & Gurevych, 2014) toolkit for processing the data and extracting features. They used both n-gram related similarity measures and a support vector machine for obtaining the results. Their approach is different from related work since they opted to view the classification rather as a ranking task, with each label being converted to to a rank between good and bad. At test time, they created a ranking of answers for each question.

In the work by Zamanov et al. (2015) a MaxEnt classifier was selected to classify question-answer pairs. The employed various stylometric and overlap-based measures, thereby focusing on machine learning modeling: they extracted a vector for each word in the question and answer, then computed the centroids of the vectors, which were then used to calculate the cosine similarity between centroids. The same procedure was also conducted with only content-bearing nominal phrases. As a framework they also used DKPro, and reported a very positive effect on accuracy when boosting the development set for training.

Rudzewitz and Ziai (2015) adapted a short answer assessment system for the task of answer selection. With their alignment-based short answer assessment system CoMiC - which also formed that basis for the present study - they aligned the question with all answers to this question under the hypothesis that relevant answers are more similar to the question. While the present study significantly outperforms their system in terms of accuracy, they gained an important insight by recognizing that a machine learning approach that takes into account the sequential thread structure of answers outperforms an approach that looks at question-answer pairs in isolation. This insight led to the implementation of the answer-answer features in the present study. The main problem that Rudzewitz and Ziai (2015) had was that their system was not adapted well enough to the web-specific language of community question answering, which was remedied in the present study via domain adaptation.

6 Discussion and Conclusion

This thesis showed that for different variants of question answering the importance of different information sources changes. With the example of community question answering it was shown that domain adaptation is beneficial for a question answering system, and that the information sources that are present in the specific domain should be made use of by the system. While for the traditional question answering task features that characterize the relation between a question and an answer are most effective, for the community question answering task domain-specific features that characterize the answer quality and its role in the discourse are more effective. These findings were obtained in a comparative experimental setting where all but the experimental factors of interest were controlled for.

This insight raises the question why the difference in strength of feature families is caused. Especially interesting is the question whether the results would change when the language in the community question answering data set would be produced in a more controlled setup where for example spelling and grammatical errors are rare. It would be interesting to see which information source would become dominant for the task of community question answering in the case where a system can't rely as much on characteristics of the answer, i.e. whether in this case the results for community question answering and traditional question answering converge. The question is whether once a comparability of the language of the question and answers is given the relation between the question is more important or the inter-answer relations. The present study can't give a definitive answer to this question due to a lack of available data, but future work needs to find an answer to it.

While the present study showed that by domain adaptation a question answering system (in this case the CoMiC system) can be improved significantly, it also has to be noted that the present study didn't push the upper bound of accuracy presented in the literature. The goal of this thesis was not to push the accuracy limit by uninterpretable models, but to gain generalizing and comparable insights in terms of information sources and domain adaptation for question answering which can be reused by future work. This goal was clearly matched.

This thesis focused on a sub challenge of question answering, namely answer selection with a given set of candidate answers. In an end-to-end system this would represent only one component in a larger system. A complete question answering system would need to gather answers first, perform the answer selection mentioned here, and output one answer based on the information found in the relevant answers.

The findings of the present study could be extended in future work in a range of different ways. While the feature families used in this study contain a representative and large (> 250) set of distinct features, there are other features that could be integrated into the analysis in future work. Since much work has been done on tree-based methods, future work should explore different tree-based methods. Also the similarity measure based on WordNet (Miller, 1995) information could be replaced by similarity methods on the basis of embeddings as obtained by distributional semantics methods. The use of word vectors would also lend itself well to neural network approaches, which have shown to yield very competitive results (Yu et al., 2014; Tan, Xiang, & Zhou, 2015). Apart from neural networks it would be worthwhile to explore other machine learning algorithms, preferably ones that are interpretable.

7 Acknowledgments

I would like to thank Ramon Ziai and Detmar Meurers for their continuous and helpful support of my work in the SFB 833 A4 project. Their ongoing support and confidence were the key factor for enabling this thesis.

References

Banerjee, S., & Lavie, A. (2005). METEOR: An automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization* (Vol. 29, pp. 65–72).

Bär, D., Zesch, T., & Gurevych, I. (2013, August). DKPro Similarity: An Open Source Framework for Text Similarity. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations* (pp. 121–126). Sofia, Bulgaria: Association for Computational Linguistics. Retrieved from http://www.aclweb.org/anthology/P13-4021

Belinkov, Y., Mohtarami, M., Cyphers, S., & Glass, J. (2015). VectorSLU: A Continuous Word Vector Approach to Answer Selection in Community Question Answering Systems. *SemEval-2015*, 282.

Bird, S. (2006). NLTK: the Natural Language Toolkit. In *Proceedings of the COLING/ACL on Interactive presentation sessions* (pp. 69–72).

Bradley, A. P. (1997). The Use of the Area under the ROC Curve in the Evaluation of Machine Learning Algorithms. *Pattern Recognition*, *30*(7), 1145–1159.

Chang, W.-t. Y. M.-W., & Pastusiak, C. M. A. (2013). Question Answering Using Enhanced Lexical Semantic Models.

Clough, P., & Stevenson, M. (2011). Developing a Corpus of Plagiarised Short Answers. *Language Resources and Evaluation*, *45*(1), 5–24.

Crossley, S. A., & McNamara, D. S. (2010). Cohesion, Coherence, and Expert Evaluations of Writing Proficiency. In *Proceedings of the 32nd annual conference of the Cognitive Science Society* (pp. 984–989).

Cui, H., Sun, R., Li, K., Kan, M.-Y., & Chua, T.-S. (2005). Question Answering Passage Retrieval using Dependency Relations. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 400–407).

Dietterich, T. G. (1998). Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. *Neural Computation*, *10*(7), 1895–1923.

Eckart de Castilho, R., & Gurevych, I. (2014, August). A Broad-Coverage Collection of Portable NLP Components for Building Shareable Analysis Pipelines. In *Proceedings of the Workshop on Open Infrastructures and Analysis Frameworks for HLT* (pp. 1–11). Dublin, Ireland: Association for Computational Linguistics and Dublin City University. Retrieved from `http://www.aclweb.org/anthology/W14-5201`

Ferrucci, D., & Lally, A. (2004). UIMA: an Architectural Approach to Unstructured Information Processing in the Corporate Research Environment. *Natural Language Engineering*, *10*(3-4), 327–348.

Gale, D., & Shapley, L. S. (1962). College Admissions and the Stability of Marriage. *The American Mathematical Monthly*, *69*(1), 9–15.

Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., … Smith, N. A. (2011). Part-of-speech Tagging for Twitter: Annotation, Features, and Experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2* (pp. 42–47).

Graesser, A. C., McNamara, D. S., Louwerse, M. M., & Cai, Z. (2004). Coh-Metrix: Analysis of Text on Cohesion and Language. *Behavior Research Methods, Instruments, & Computers*, *36*(2), 193–202.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA Data Mining Software: An Update. *ACM SIGKDD explorations newsletter*, *11*(1), 10–18.

Heilman, M., & Smith, N. A. (2010). Tree Edit Models for Recognizing Textual Entailments, Paraphrases, and Answers to Questions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics* (pp. 1011–1019).

Hosmer, D. W., & Lemeshow, S. (2000). Introduction to the Logistic Regression Model. *Applied Logistic Regression, Second Edition*, 1–30.

Hou, Y., Tan, C., Wang, X., Zhang, Y., Xu, J., & Chen, Q. (2015). HITSZI-CRC: Exploiting Classification Approach for Answer selection in Community Question Answering. In *Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval* (Vol. 15, pp. 196–202).

Krifka, M. (2008). Basic Notions of Information Structure. *Acta Linguistica Hungarica*, *55*(3-4), 243–276.

Levenshtein, V. I. (1966). Binary Codes Capable of Correcting Deletions, Insertions and Reversals. In *Soviet physics doklady* (Vol. 10, p. 707).

Li, X., & Roth, D. (2002). Learning Question Classifiers. In *Proceedings of the 19th International Conference on Computational Linguistics-Volume 1* (pp. 1–7).

Lin, X. Z. B. H. J., & Wang, Y. X. X. (2015). ICRC-HIT: A Deep Learning based Comment Sequence Labeling System for Answer Selection Challenge. *SemEval-2015*, 210.

Manning, C. D., Raghavan, P., Schütze, H., et al. (2008). *Introduction to Information Retrieval* (Vol. 1) (No. 1). Cambridge University Press Cambridge.

Màrquez, L., Glass, J., Magdy, W., Moschitti, A., Nakov, P., & Randeree, B. (2015). Semeval-2015 Task 3: Answer Selection in Community Question Answering. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015).*

McCallum, A. K. (2002). *MALLET: A Machine Learning for Language Toolkit.* (http://mallet.cs.umass.edu)

McCullagh, P., & Nelder, J. A. (1989). *Generalized Linear Models* (Vol. 37). CRC press.

Meurers, D., Ziai, R., Ott, N., & Bailey, S. M. (2011). Integrating Parallel Analysis Modules to evaluate the Meaning of Answers to Reading Comprehension Questions. *International Journal of Continuing Engineering Education and Life Long Learning*, *21*(4), 355–369.

Meurers, D., Ziai, R., Ott, N., & Kopp, J. (2011). Evaluating Answers to Reading Comprehension Questions in Context: Results for German and the Role of Information Structure. In *Proceedings of the TextInfer 2011 Workshop on Textual Entailment* (pp. 1–9).

Meysam, A. H. A. S. R., Fakhrahmad, R. M., & Street, E. (2015). Shiraz: A Proposed List Wise Approach to Answer Validation. *SemEval-2015*, 220.

Miller, G. A. (1995). WordNet: a lexical database for English. *Communications of the ACM*, *38*(11), 39–41.

Myers, E. W. (1986). An O(ND) difference algorithm and its variations. *Algorithmica*, *1*(1-4), 251–266.

Nakov, P., Màrquez, L., Moschitti, A., Magdy, W., Mubarak, H., Freihat, A. A., ... Randeree, B. (2016, June). SemEval-2016 Task 3: Community Question Answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation.* San Diego, California: Association for Computational Linguistics.

Nicosia, M., Filice, S., Barrón-Cedeno, A., Saleh, I., Mubarak, H., Gao, W., ... others (2015). QCRI: Answer Selection for Community Question

Answering Experiments for Arabic and English. In *Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval* (Vol. 15, pp. 203–209).

Nivre, J., Hall, J., Nilsson, J., Chanev, A., Eryigit, G., Kübler, S., . . . Marsi, E. (2007). MaltParser: A language-independent System for data-driven Dependency Parsing. *Natural Language Engineering*, *13*(02), 95–135.

Peng, C.-Y. J., Lee, K. L., & Ingersoll, G. M. (2002). An Introduction to Logistic Regression Analysis and Reporting. *The Journal of Educational Research*, *96*(1), 3–14.

Punyakanok, V., Roth, D., & Yih, W.-t. (2004). Mapping Dependencies Trees: An Application to Question Answering. In *Proceedings of AI&Math 2004* (pp. 1–10).

Renyi, A. (1961). On measures of Entropy and Information. In *Fourth Berkeley symposium on mathematical statistics and probability* (Vol. 1, pp. 547–561).

Rudzewitz, B. (2016, June). Exploring the Intersection of Short Answer Assessment, Authorship Attribution, and Plagiarism Detection. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications* (pp. 235–241). San Diego, CA: Association for Computational Linguistics. Retrieved from `http://www.aclweb.org/anthology/W16-0527`

Rudzewitz, B., & Ziai, R. (2015). CoMiC: Adapting a Short Answer Assessment System for Answer Selection. In *Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval* (Vol. 15).

Santos, C. d., Tan, M., Xiang, B., & Zhou, B. (2016). Attentive Pooling Networks. *arXiv preprint arXiv:1602.03609*.

Schmid, H. (2013). Probabilistic Part-of-Speech Tagging using Decision Trees. In *New Methods in Language Processing* (p. 154).

Severyn, A., & Moschitti, A. (2013). Automatic Feature Engineering for Answer Selection and Extraction. In *EMNLP* (pp. 458–467).

Severyn, A., & Moschitti, A. (2015). Learning to rank Short Text Pairs with Convolutional Deep Neural Networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 373–382).

Stamatatos, E. (2009). A survey of modern authorship attribution methods. *Journal of the American Society for information Science and Technology*, *60*(3), 538–556.

Tan, M., Xiang, B., & Zhou, B. (2015). LSTM-based Deep Learning Models for non-factoid Answer Selection. *arXiv preprint arXiv:1511.04108*.

Tran, Q. H., Tran, V., Vu, T., Nguyen, M., & Pham, S. B. (2015). JAIST: Combining Multiple Features for Answer Selection in Community Question Answering. In *Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval* (Vol. 15, pp. 215–219).

Vajjala, S., & Meurers, D. (2012). On Improving the Accuracy of Readability Classification using Insights from Second Language Acquisition. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP* (pp. 163–173).

Vo, N. P. A., Magnolini, S., & Popescu, O. (2015). FBK-HLT: An Application of Semantic Textual Similarity for Answer Selection in Community Question Answering. *SemEval-2015*, 231.

Wang, D., & Nyberg, E. (2015). A Long Short-Term Memory Model for Answer Sentence Selection in Question Answering. *ACL, July*.

Wang, M., & Manning, C. D. (2010). Probabilistic Tree-Edit Models with Structured Latent Variables for Textual Entailment and Question Answering. In *Proceedings of the 23rd International Conference on Computational Linguistics* (pp. 1164–1172).

Wang, M., Smith, N. A., & Mitamura, T. (2007). What is the Jeopardy Model? A Quasi-Synchronous Grammar for QA. In *EMNLP-CoNLL* (Vol. 7, pp. 22–32).

Wang, Z., & Ittycheriah, A. (2015). FAQ-based Question Answering via Word Alignment. *arXiv preprint arXiv:1507.02628*.

Wang, Z., Mi, H., & Ittycheriah, A. (2016). Sentence Similarity Learning by Lexical Decomposition and Composition. *arXiv preprint arXiv:1602.07019*.

Yao, X., Van Durme, B., Callison-Burch, C., & Clark, P. (2013). Answer Extraction as Sequence Tagging with Tree Edit Distance. In *HLT-NAACL* (pp. 858–867).

Yi, L., Wang, J., & Lan, M. (2015). ECNU: Using Multiple Sources of CQA-based Information for Answer Selection and YES/NO Response Inference. *SemEval-2015*, 236.

Yu, L., Hermann, K. M., Blunsom, P., & Pulman, S. (2014). Deep Learning for Answer Sentence Selection. *arXiv preprint arXiv:1412.1632*.

Zamanov, I., Hateva, N., Kraeva, M., Yovcheva, I., Nikolova, I., & Angelova, G. (2015). Voltron: A Hybrid System for Answer Validation based on Lexical and Distance Features. *SemEval-2015*, 242.