A Customizable Grammar-Based Framework for User-Intent Text Classification

Thesis by

ALAA MOHASSEB

This thesis is submitted in partial fulfilment of the requirements for the award of the degree of Doctor of Philosophy of the University of Portsmouth

July 2018

To the memory of my mother, Hamida.

"You have always been an outstanding inspiration to me. Getting to this stage in my life has taken a lot of work, but it is nothing compared to how you worked and sacrificed for me. You are the number one reason I am where I am today. Without your continued support, I could never have accomplished so much. I love you."

Acknowledgments

There are no words that describe how grateful I am to the people who supported me during this *"bittersweet"* journey.

First of all, I would like to thank my supervisor Doctor *Mohamed Bader* for his constant guidance, great support and kind advice throughout my PhD research studies. It was a real privilege and an honour for me to be his student. I am thankful for his enthusiasm and immense knowledge which without I would not get to where I am now. I am also grateful for his extraordinary human qualities and for providing me with so many wonderful opportunities.

I also would like to thank my co-supervisor Doctor *Mihaela Cocea* for her constant support, availability and constructive suggestions, which helped in the accomplishment of the work presented in this thesis. I am thankful for her friendship, mentorship and for being someone to look up to.

To my Father, *Ali*; thank you so much for everything! I am grateful for your help to get a great education in these past years. I would never have been able to succeed without you. To my siblings, *Ahmed* and *Amani*; thank you for always supporting and encouraging me.

I also would like to thank my *colleagues* and *friends*, in the School of Computing, for their companionship and for providing a pleasant and friendly working atmosphere. Finally, a special thanks go to my friend *Nessrin*, for always being there for me and for helping me overcome many difficult moments. I am blessed to have a friend like you.

Declaration

Whilst registered as a candidate for the above degree, I have not been registered for any other research award. The results and conclusions embodied in this thesis are the work of the named candidate and have not been submitted for any other academic award.

Abstract

In real-life classification problems, prior information about the problem and expert knowledge about the domain are often used to obtain reliable and consistent solutions. This is especially true in fields where the data is ambiguous, such as text, in which the same words can be used in seemingly similar texts but have a different meaning. Many of the proposed approaches rely on the bag-of-words representation, which loses the information about the structure of the text. In this thesis, a literature review of related works in text classification is provided which includes an overview of text classification methods. In addition, detailed review of related works of two text classification domains; search engines and question answering systems. The core contribution is divided into three main parts. The first contribution is the Customizable Grammar Framework for user-intent text classification (CGF) which employs a formal grammar approach and exploits domain-related information in a new way to represent text as a series of syntactic categories forming syntactic patterns. In addition, the proposed framework has been applied to different domains which resulted in the second and third contribution. The second contribution is the Grammar-Based Framework for Query Classification (GQC) which helped in the improvement of query identification and classification. The third contribution is the Grammar-Based Framework for Question Categorization and Classification (GQCC) which helped in the enhancement of question identification and classification. In addition, using different machine learning algorithms the overall results show that the proposed approach outperforms previous ones in terms of classification performance for query and question classifications. Finally, comparison of the classification performance with the state-of-the-art approaches has been conducted, results validate that the proposed approach improves the classification accuracy and the identification of the different types of queries and questions.

Contents

List of Figures

List of Tables

Acronyms

Acc Accuracy.

Alg Algorithms.

ANN Artificial Neural Networks.

BNF Backus Normal Form.

BoW bag-of-words.

CFG Context-Free Grammar.

CFSG Context-Free Multiset Generating Grammar.

CGF Customizable Grammar Framework.

CNN Convolutional Neural Networks.

CQA Community Question Answering.

CRFs Conditional Random Fields.

CRNN Context-Sensitive Recursive Neural Network.

CVG Compositional Vector Grammar.

DT Decision Tree.

GQC Grammar Based Framework for Query Classification.

GQCC Grammar Based Framework for Question Categorization and Classification.

HBKNN Hybrid KNN.

HINs Heterogeneous Information Networks.

HMM Hidden Markov Models.

Idf Inverse Frequency in Documents.

IR Information Retrieval.

KNN k-Nearest Neighbor.

LSI Latent Semantic Indexing.

LSTM Long short-term memory.

ME Maximum Entropy.

NB Naïve Bayes.

NLP Natural Language Processing.

NN Neural Networks.

P Precision.

PCFG Probabilistic Context-Free Grammar.

PoS Part-of-Speech.

QASs Question-Answering Systems.

QC Questions Classification.

R Recall.

RF Random Forests.

RNN Recurrent Neural Networks.

SFG Systemic Functional Grammar.

SMOTE Synthetic Minority Over-sampling Technique.

SNoW Sparse Network of Winnows.

SVM Support Vector Machine.

SVMT Support Vector Machines Tagger.

TBL Transition-Based Learning.

TF Term Frequency.

TREC Text REtrieval Conference.

VSM Vector Space Model.

WSJ Wall Street Journal.

Publications

Journal Papers

1) Mohasseb, A., Bader-El-Den, M., Cocea, M.: Question categorization and classification using grammar based approach. Information Processing & Management (2018)

Conference Papers

1) Mohasseb, A., Bader-El-Den, M., Cocea, M.: Detecting question intention using a k-nearest neighbor based approach. In: IFIP International Conference on Artificial Intelligence Applications and Innovations. pp. 101–111. Springer (2018)

2) Mohasseb, A., Bader-El-Den, M., Cocea, M.: Analysis of the syntactical structure of web queries. In: Machine Learning and Cybernetics (ICMLC), 2018 International Conference on. IEEE (2018)

3) Mohasseb, A., Bader-El-Den, M., Cocea, M., Liu, H.: Improving imbalanced question classification using structured smote based approach. In: Machine Learning and Cybernetics (ICMLC), 2018 International Conference on. IEEE (2018)

4) Mohasseb, A., Bader-El-Den, M., Kanavos, A., Cocea, M.: Web queries classification based on the syntactical patterns of search types. In: International Conference on Speech and Computer. pp. 809–819. Springer (2017)

5) Mohasseb, A., Bader-El-Den, M., Liu, H., Cocea, M.: Domain specific syntax based approach for text classification in machine learning context. In: Machine Learning and Cybernetics (ICMLC), 2017 International Conference on. vol. 2, pp. 658–663. IEEE (2017)

CHAPTER 1

Introduction

With the increasing size and diversity of online data, the field of Information Retrieval (IR) is continually evolving. The process of searching and obtaining information relevant to the information needed is increasingly challenging in terms of processing and analyzing it.

A user's need, which can initially be vague, is expressed in the form of a request; this request could be any kind of information. This information takes the form of a digital text and could be website articles, research papers and blog entries. The challenge is to provide a good match between the user's given information (e.g. text) and the user's need in order to ensure that the retrieved information is relevant. As a result, many approaches from different IR applications have become very important such as web search engines and question answering systems. These applications enabled users to interactively search for relevant information.

For example, web search engines which are the most popular information retrieval applications such as Google and Yahoo! became an integral part of people's lives but despite the fact that they try to improve the user experience and the technology used in finding relevant results, the results returned by search engines are still overwhelming to most users.

Another example of a popular information retrieval application is question answering systems such as answers.com, the usage of question answering systems is increasing daily as people constantly use them in order to find the right answer for different kinds of information. However, similar to search engines the results returned could be overwhelming to most users. Results are highly sensitive to vocabulary, due to the difficulty in understanding the contextual meaning of the terms. Cases of polysemy like *("Apple" as a fruit vs. "Apple" as a company)* and cases of synonymy like *("movies" and "films")* could lead to ambiguity and retrieval of irrelevant information.

The main challenge facing the improvement of such results is the difficulty of classifying

and determining the user's intent. One major task in the improvement of such results is identifying the intent and the accurate classification; therefore, the analysis of data contained in a text is one important step through the process of structuring the input text, using techniques such as parsing, tagging and linguistic features.

The following sections are organized as follow: section 1.1 presents the thesis motivation, section 1.2 describes in detail the aim and objectives, and Section 1.3 provides an overview of the thesis.

1.1 Motivation

In real-life classification problems, some prior information about the structure of the problem are known in advance, such as the relation between some attributes or the patterns that are likely to appear in certain instances. Moreover, the features extracted from many real-world problems are not completely independent and the meaning of each feature may be influenced by other attributes and/or the position of the attribute in the instance. This is especially true in fields where the data is ambiguous, such as text, in which the same words can be used in seemingly similar texts, but have different meaning; in addition to words in the text, the syntax plays an important role in defining the meaning of the text.

Nevertheless, the performance of text classifiers highly depends on the problem domain, as it is unlikely to find a single classifier that outperforms all other classifiers on all domains, leading to approaches that take domain information into account. In order to achieve highly accurate classification models, the development of configurable classifiers, that could be customized to a given domain is crucial.

Most information retrieval solutions are based on bag-of-words and dictionaries/lexicons representation which looses the information about the structure of the text. A limitation of these approaches is that the meaning of words or groups of words (called terms) which could be one or more words is ambiguous. For example, "Order Danielle Steel books" and "Danielle Steel books order" consist of very similar terms but reflect different intentions, the intent of the first example is *to buy Danielle Steel books*, while the intent of the second example is *find information on Danielle Steel books*.

One of the most researched areas within text classification is query classification, which has emerged as an area of research aiming to improve the relevance of retrieved information

by classifying queries according to the users' needs. While many approaches focused on identifying the topic (e.g. news, sports, hotels) the user was interested in, other approaches focused on user intent, i.e. the purpose of the search.

Another popular researched areas within text classification is question classification, which similar to query classification it's an area of research aiming to improve the relevance of retrieved information/answers by classifying questions according to the users' needs. Although many approaches focused on user intent, i.e. the type of the question, there are other approaches that focused on identifying the question topic (e.g. celebrity, sports, pets).

To address the limitation of these domains and classification tasks, a classification framework that aims to focus on user-intent text classification is proposed. The following section outlines in detail the aim and objectives of the thesis and how they will be addressed.

1.2 Aim and Objectives

To address the limitation of word/term-based approaches that typically ignore the order and relations between terms within a piece of text, a framework was proposed for the classification that exploits the structure of the text, thus preserving both order and term relations in which the proposed approach addresses one of the major issues in text representation, i.e. large sparse datasets, by requiring a significantly smaller number of features. Furthermore, the use of formal grammatical rules as a method was investigated to capture domain specific information and the structure of the text by transforming the text into a new representation of syntactic categories and patterns. In addition, assess the influence of using the structure of a text and the domain-specific syntactic categories on the classification performance. To achieve this aim, the following objectives are defined:

1) Identify the state-of-the-art methods and research gaps in the area of text classification.

2) Design a framework for general user-intent text classification.

3) Apply the proposed framework on different domains and classification tasks.

4) Evaluate the impact of using different levels of detail of syntactic categories and domain-specific information on the classification performance and compare the classification performance of different machine learning algorithms.

5) Evaluate the classification performance in comparison with state-of-the-art approaches.

By addressing the above objectives, the thesis makes the following contributions:

A Customizable Grammar-Based Framework (CGF) for user intent text classification is proposed in chapter 3 to address the limitations of general approaches in text classification and incorporate domain-related information without increasing the complexity of the textual representation and computation, as well as take into account the structure of the text. CGF has the following novel features: (a) the text is represented as a syntactic pattern, i.e. each term is replaced by its corresponding syntactic category and all syntactic categories in the piece of text form the syntactic pattern; (b) the syntactic categories used are not just the standard English ones, but also domain-specific syntactic categories; (c) a formal grammar approach is used to transform a piece of text into a syntactic pattern. Machine learning is applied on this transformed data to obtain models for automatic classification. In addition, *grammatical rules and patterns* were created which helped in improving terms ambiguity and the identification of different terms. A detailed explanation of these rules and patterns is provided in chapter 3. Furthermore, *a syntax based parsing and tagging* is developed for the objective of assigning not just Part-of-Speech (PoS) tags but also domain specific ones to help in the categorization and classification of text in different domains. The parsing and tagging approach is presented in chapter 3.

Moreover, the framework is applied to query classification and question classification problems, in which *A Grammar Based Framework for Query Classification (GQC)* was adapted from CGF. GQC is created and modified in a way that helped to improve query identification and classification. A full description of this framework is provided in chapter 4. In addition, *A Grammar Based Framework for Question Categorization and Classification (GQCC)* is introduced which was also adapted from CGF and adjusted in a way that helped in the enhancement of question identification, categorization and classification. A full description of this framework is provided in chapter 5.

1.3 Thesis Overview

The rest of the thesis is organized as follows: Chapter 2 presents an overview of related work, highlighting the different methods and approaches that have been used in text classification in general, and web search query classification and question classification in particular.

Chapter 3 introduces the customizable grammar-based framework for user intent text classification which address the limitations of general approaches in text classification by exploiting domain-related information and present the text as a series of syntactic categories forming syntactic patterns. In Chapter 4, the grammar-based framework for query classification is discussed which helped in the identification of different query types based on the identified syntactic categories and the formal grammar. Following this, the grammar-based framework for question categorization and classification is presented in Chapter 5, by creating domain specific grammatical rules and patterns for each type of question. Last, conclusions drawn from the work along with directions for future work are presented in Chapter 6.

CHAPTER 2

Literature Review

In order to identify the user-intent behind any search whether it is a general query or a question, one of the objectives is to understand the meaning behind the search. This chapter represents the research efforts made towards that objective. First, a literature review of related works in text classification is presented in Section 2.1 in which an overview of text classification methods and techniques is outlined such as features and machine learning based approaches. Following that, Section 2.2 describes text parsing and tagging approaches and the role it plays as one of the fundamental phases in text processing. Sections 2.3 presents one of the classification tasks that is presented in this thesis which is query classification, in which the different categories that helped in the analysis and understanding of search engine user intent is outlined, in addition to the different methods and approaches that have been proposed in order to improve the understanding and classification of users' search queries. Sections 2.4 presents the second classification task that is presented in this thesis which is question classification. This section outlines the different categories of questions that have been proposed and the different methods and approaches that have been used in order to improve the understanding and classification of users' questions. Finally, Section 2.5 summarizes the chapter and presents a discussion of the main observations drawn from previous work that motivated the grammar-based approach proposed in this thesis for user-intent text classification.

2.1 Text Classification

Text classification which could be defined as the task of labeling natural language texts with thematic categories from a predefined set [124], is an important task in Natural Language Processing with many applications, such as web search (e.g. [47], [150], [43]), question-answering (e.g. [159], [41], [74]), sentiment analysis (e.g. [1], [135], [32], [152]). However,

traditional text classifiers often rely on many human-designed features, such as dictionaries, knowledge bases and special tree kernels rather than the relations between the entities, as well as the types of the entities and relations which carry much more information to represent the texts [147].

The selection of distinctive features is essential for text classification [143] [142]. A key problem in text classification is feature representation, which is commonly based on the bag-of-words (BoW) model, where uni-grams, bi-grams, n-grams or some exquisitely designed patterns are typically extracted as features [62].

2.1.1 Text Classification Methods and Techniques

In the following sub-sections, a detailed review on text classification is presented and the different types of methods and techniques.

2.1.1.1 Features and Machine Learning-Based Approaches

Many different machine learning approaches have been used to classify natural language sentences and words; recurrent neural networks is one of the approaches that have been used by many researches. In [63] and [117] a Recurrent Neural Networks (RNN) were used to classify natural language sentences as grammatical or ungrammatical. In [117] encoded natural language sentences were used as examples to train a recurrent neural network; this encoding was based on the linguistic theory of Government and Binding [20]. Authors in [63] also examined the use of various recurrent neural network architectures like FGS, N&P, Elman, and W&Z to train a network for classification.

Authors in [62] introduced a recurrent convolutional neural network for text classification without human-designed features. A recurrent structure is applied to capture contextual information when learning word representations. A max-pooling layer were also employed that automatically judges which words play key roles in text classification to capture the key components in texts.

Furthermore, [148] proposed a method to model short texts based on semantic clustering and convolutional neural network. First semantic cliques are discover in embedding spaces by a fast clustering algorithm. Then, multi-scale semantic units are detected under the supervision of semantic cliques, which introduce useful external knowledge for short texts. These

meaningful semantic units are combined and fed into the convolutional layer, followed by max-pooling operation. Experimental results were conducted on two open benchmarks. The results validated the effectiveness of this method.

Works in [22] presented a new architecture Deep Convolutional Neural Networks (VD-CNN) for text processing which operates directly at the character level and uses only small convolutions and pooling operations. This architecture has been evaluated on eight freely available large-scale data sets and the performance of this model increases with the depth using up to 29 convolutional layers. Experiments showed an improvement over the state-of-the-art on several public text classification tasks.

According to [76] most previous neural network based methods are learnt based on single-task supervised objectives, which often suffer from insufficient training data. To jointly learn across multiple related tasks based on recurrent neural network a multi-task learning framework was used. Moreover, a three RNN based architectures were used to model text sequence with multi-task learning of sharing information to model text with task-specific and shared layers in which the entire network is trained jointly on all these tasks.

Other researches used machine learning algorithms such as K-Nearest Neighbour as a mean of classification, in addition to feature selection. Authors in [5] stated that automatic feature selection methods are extremely important to handle the high dimensionality of data for effective text classification, so a new supervised feature selection approach was proposed to improve the performance of text classification which develops a similarity between a term and a class.

Works in [104] proposed a mining model consists of sentence, document and corpus-based concept-analysis. The term that contributes to the sentence semantics was analyzed on the sentence, document, and corpus levels rather than the traditional analysis of the document only. After extracting feature vector for each new document, feature selection was performed. It is then followed by K-Nearest Neighbour classification.

While in [75] a method was proposed that combined clustering and feature selection to labels set of representative words for each class then uses these words to extract a set of documents for each class from a set of unlabelled documents to form the initial training set. Expectation–Maximization (EM) algorithm is applied to build the classifier. This technique can effectively rank the words in the unlabelled set according to their importance and the user then selects/labels some words from the ranked list for each class.

Furthermore, authors in [157] stated that sparse, imbalance, and noise are some of the limitations that Conventional k-nearest Neighbor (KNN) classification approaches have when dealing with some special datasets. They designed RS-HBKNN classifier in order to improve the performance of hybrid KNN (HBKNN). In [149] authors implemented a text classification system based on mutual information and K-nearest neighbour algorithm and support vector machine.

Naive Bayes has also been used to automatically classify text but according to the authors in [55] while naive Bayes is effective in various data mining tasks, it shows a disappointing result in the automatic text classification problem. They stated that naive Bayes for the natural language text, has a serious problem in the parameter estimation process, which causes poor results in the text classification domain. Two empirical heuristics were proposed; per-document text normalization and feature weighting method. The proposed naive Bayes text classifier performed very well in the standard benchmark collections, competing with state-of-the-art text classifiers based on a highly complex learning method such as SVM.

In [81] authors proposed a method based on WordNet thesaurus and Latent Semantic Indexing (LSI) model to realize Naive Bayes text classification and simple vector distance text classification. According to them incorporating linguistic knowledge into the text representation can lead to improvements in classification accuracy.

Authors in [38] introduced a learning algorithm to classify documents from fully unlabeled documents based on the combination of a Naive Bayes classifier and expectation-maximization using class associated words; to set classification constraints class associated words are used during the learning process to classify documents into equivalent class labels and improve the classification accuracy. Finally, works in [33] designed a web of Chinese text categorization system model and system tested based on the Bayes theory.

2.1.1.2 Other Text Classification Approaches

Some other approaches have been used for classification like knowledge tree, multilayer and n-gram. [108] stated that most researchers focus on statistic method like (Rocchio, SVM, KNN) which is based on Vector Space Model (VSM) representing text, so they introduced a new method for automatic text classification based on knowledge tree to simulate the process of human classification and it included background knowledge and classification algorithm. This algorithm is based on text semantic structure to avoid the disadvantages of SVM. It

combined text semantic structure and background knowledge to activate relative branches of knowledge tree and decide which classification it belongs to by reasoning.

A text classification was proposed in [132] which is based on multilayer SVM-NN text classification and two-level representation model; one is for representing syntactic information using tf-idf value and the other is for semantic information using Wikipedia. Furthermore, a multi-layer text classification framework is designed to make use of the semantic and syntactic information. The proposed framework contains three SVM-NN classifiers in which two classifiers are applied on syntactic level and semantic level in parallel. The outputs of these two classifiers were then combined and given as input to the third classifier.

Moreover, [158] introduced a method to discriminatively learn phrase patterns to be used as features in text classification; they used a recursive algorithm with a mutual information selection criterion to search for phrase patterns and the upper-bound of the mutual information is used to terminate the search early; the computation of the upper-bound requires only the statistics of the prefix pattern. The specific locations of a phrase pattern is automatically determined when word classes are useful, allowing for variable specificity depending on the amount of labelled data available.

According to [160] KNN is sensitive to the distance or similarity. A function has been used in classifying a test instance which can cause low classification accuracy and limit the KNN classifier's utilization in text classification in text mining. A mahalanobis distance in text classification area was introduced; in addition, an algorithm (MDKNN) base on this theory was proposed.

Finally, authors in [147] proposed a novel text as network classification framework, which is based on a structured and typed Heterogeneous Information Networks (HINs) representation of texts, and a meta-path based approach to link texts, this new representation and links of texts, can be incorporated into kernels. In addition, a SVM classifier was developed using indefinite kernel matrices based on KnowSim [146], which is a knowledge driven text similarity measure that could naturally encode the structural information in the text HINs. Experiments were conducted on two benchmark datasets which showed that the indefinite HIN-kernel based on weighted meta-paths outperforms the state-of-the-art methods and other HIN-kernels.

2.2 Text Parsing and Tagging

Parts-of-Speech (PoS) which could be defined as a category to which a word is assigned in accordance with its syntactic functions; provides large amounts of information about a word. It plays an important role in Natural Language Processing (NLP). In English the main parts-of-speech are noun, pronoun, adjective, determiner, verb, adverb, preposition, conjunction, and interjection. Knowing whether a word is a noun or a verb helps in identifying other words in the same sentence e.g. nouns are preceded by determiners and adjectives while verbs are preceded by nouns. In addition, it helps in identifying the syntactic structure around the word e.g. nouns are generally part of noun phrases. Furthermore, PoS are useful features for finding named entities like people or organizations in text and other information extraction tasks, which makes part-of-speech tagging an important component of syntactic parsing.

PoS parsing and tagging is one of the fundamental phases in text processing. Parsing has been used as a way to identify the sentence structure by adding mark ups which helps in organizing a sentence while tagging represent classes and features of terms in which each word will receive a tag based upon its word class and the feature it holds.

A broad range of PoS parsing and tagging tools and approaches have been developed; most of these tools and approaches are based on natural language which focus mostly on the development of statistical parsers and tagging in which most of them are trained on large annotated corpora in newswire domain like the Penn Treebank WSJ corpus [83]. According to [136] some statistical parsers have shown good results on this benchmark but when they are applied to data from different domain they have demonstrated worse results [118], [28], [114]. Furthermore, parsers and taggers still suffer from the problem of domain adaptation [111], [85], since most of them are based just on NLP tags which cannot be used in domains such as search engines, question answering systems and social networks; knowing only the PoS tag will not assist in identifying and retrieving relevant information since most of these domains are not based only on PoS grammar. For example, web queries usually do not follow the formal English grammar like word order, and no labelled syntactic trees for web queries are available. Moreover, using queries on parsers that are trained on standard treebanks leads to poor results [133]. Similar to web queries, Twitter poses additional challenges due to the conversational nature of the text and the lack of conventional orthography, and the 140-character limit of each message (“tweet”) [31].

Recent studies have used different features to assist the parsing and tagging process like Hidden Markov Models (HMM) [122], [10], [65]; other works like [105] and [136] used PoS and Lexical Features. In addition, some works used grammar for this process, authors in [82] proposed a Context-Free Multiset Generating Grammar (CFSG), while [57] proposed unlexicalized Probabilistic Context-Free Grammar (PCFG) and authors in [126] introduced a Compositional Vector Grammar (CVG). Furthermore, many previous studies used different machine learning algorithms. Support Vector Machine (SVM), Neural Networks (NN) and Maximum Entropy (ME) are the most used algorithms. Combining a classifier with different features such as semantic, syntactic and lexical improves the parsing and tagging process. Authors in [29], [30] and [82] used SVM, while [17], [10], [115] and [138] used ME and [19], [60] and [133] used NN and RNN. Other works like [136] combined both SVM and NN.

In the following sub-sections, a detailed review on parsing and tagging is presented. In addition, different methods of parsing are outlined in Section 2.2.1, while previous works on tagging methods are outlined in Section 2.2.2 .

2.2.1 Parsing

In the following sub-sections, a detailed review on parsing is presented in addition to the different types of parsers.

2.2.1.1 Grammar-Based Parsers

There are many different parsing methods and models that have been proposed. Some of these are grammar-based parsers, which focus on a certain type of formal grammar and a set of production rules. Authors in [6] proposed a general framework for maximum-margin training of context-free grammars parsers based on structural SVM.

Authors in [126] introduced a Compositional Vector Grammar (CVG), which combines Probabilistic Context-Free Grammar (PCFG) with the semantic richness of neural word representations and compositional phrase vectors. The compositional vectors are learned with a new syntactically untied recursive neural network. The CVG improves the PCFG of the Stanford parser by 3.8% to obtain an F1 score of 90.4%. According to the authors, it is fast to train and implement as an efficient re-ranker. Approximately, it is about 20% faster than

the current Stanford factored parser. The CVG learns a soft notion of head words and improves performance on the types of ambiguities that require semantic information such as PP attachments.

In [57] an unlexicalized Probabilistic Context-Free Grammar (PCFG) was proposed, this grammar make use of a linguistically motivated state splits, which break down false independence assumptions latent in a vanilla treebank grammar. This approach achieved an accuracy of 86.36% which consider better than the early lexicalized PCFG models.

Furthermore, authors in [17] proposed a lexicalized Markov grammar parsing model for parsing down to Penn tree-bank style parse trees, this approach is based on the Maximum-Entropy machine learning algorithm which helped in testing and combining many different conditioning events. Experiments showed that the parser achieved an average precision/recall of 91.1% on sentences of length < 40 and 89.5% on sentences of length < 100.

2.2.1.2 Dependency-Based Parsers

Many studies are based on the dependency parsing, which is a parser that analyzes the grammatical structure of a sentence, establishing relationships between ”head” words and words which modify those heads [19]. Authors in [111] developed distant-supervised algorithms that use dependency grammar. The proposed algorithms do not require manually parsed queries for training. Instead, millions of (query, page title) pairs from the Community Question Answering CQA domain were used to train the algorithms. Experimental results showed that the algorithms outperform other baselines.

In [161] authors developed a graph-based and a transition-based projective dependency parser using beam-search, using discriminative perceptron training and beam-search decoding these two parsers were combined into a single system. Experiments showed that the proposed parsers outperformed the pure graph-based and the pure transition-based parsers. In addition, these parsers have been tested on the English and Chinese Penn Treebank data, achieving accuracy of of 92.1% and 86.2%, respectively.

Furthermore, in [59] authors presented a simple semi-supervised method for training dependency parsers. The proposed method focuses on the problem of lexical representation and uses features that combine word clusters which are derived from a large unannotated corpus. Using the Penn Treebank and Prague Dependency Treebank, experimental study showed that the cluster-based features improved the performance across a wide range of conditions, such

as the case of English unlabelled second-order parsing, in which the baseline accuracy improved from 92.02% to 93.16%, and in the case of Czech unlabelled second-order parsing, the baseline accuracy improved from 86.13% to 87.13%.

Authors in [24] proposed a system that extract typed dependency parses of English sentences from phrase structure parses. The typed dependencies represents dependencies between individual words and labels dependencies with grammatical relations, such as subject or indirect object while phrase structures represents nesting of multi-word constituents. The system was evaluated on a sample of 10 sentences, achieving an accuracy of 80.3% per-dependency. While in [105] authors introduced a data-driven based parser generator for dependency parsing called MaltParser. The proposed parser can be used to create a parser for a new language given a dependency treebank representing that language in which this approach allows the user to choose between different parsing algorithms and learning algorithms and to define different feature models such as lexical features, part-of-speech features and dependency type features. In addition, In [106] using data from ten different languages; experimental evaluation showed that MaltParser has achieved a good parsing accuracy without language-specific enhancements and with limited amounts of training data.

Some works like [19], [133] and [130] used machine learning algorithms such as Neural Network (NN) and Recursive Neural Networks (RNN) with their dependency parser. Authors in [19] combined a dependency parser with neural networks, the proposed approach learns and uses a small number of dense features. Experimental evaluations showed that when all words, PoS tags and arc labels are represented as dense vectors, the parser achieved about 2% improvement in unlabelled and labelled attachment scores on both English and Chinese datasets.

While authors in [133] introduced algorithms to derive a query's syntactic structure from the dependency trees of its clicked sentences after acquiring well-formed sentences that contain the semantics of the query, and then infer the syntax of the query from the sentences. This creates a treebank for web queries, then a neural network dependency parser is trained from the treebank. Experiments showed that the proposed algorithms achieved significant improvement over traditional parsers on web queries.

Furthermore, in [130] authors proposed a general compositional vector framework for transition-based dependency parsing. In addition, they introduced the concept of a Transition Directed Acyclic Graph that allowed them to apply Recursive Neural Networks for parsing

with existing transition-based algorithms. The proposed framework captures semantic relatedness between phrases similarly to a constituency-based counterpart from the literature syntactically different phrases expressing financial trading. The proposed framework achieved 86.25% in unlabelled attachment score for a well-established dependency dataset using only word representations as input, falling less than 2% points short of a previously proposed comparable feature-based model.

2.2.1.3 Semantic-Based Parsers

Works like [60], [140] and [140] introduced a semantic based parser model, which could be defined as the task of converting a natural language statement to a logical form and a machine-understandable representation of its meaning [50]. In [60] authors presented a semantic parsing model for answering compositional questions on semi-structured Wikipedia tables. The proposed model extends the recent neural semantic parsers by enforcing type constraints during logical form generation, and by including an explicit entity embedding and linking module that enables it to identify entity mentions while generalizing across tables. In addition, the parser is an encoder-decoder neural network that combines a grammar for the decoder that only generates well-typed logical forms; and an entity embedding and linking module that identifies entity mentions while generalizing across tables. Furthermore, another method was proposed for training the neural model with question-answer supervision. Experiments showed that the parser has achieved accuracy of 43.3% for a single model and 45.9% for a 5-model ensemble On the WIKITABLEQUESTIONS data set.

Moreover, authors in [140] and [156] used machine learning algorithms such as Neural Network (NN) and convolutional Neural Networks (CNN). In [140] authors presented an initial study towards bringing together the semantic web experience and statistical natural language semantic parsing modeling. This study mined search queries hitting the structured web pages to semantically annotate them and built statistical unsupervised slot filling models. Furthermore, results are presented using a natural-language-like query set and a control test set for assessing the performance of the models. In addition, MAP adaptation is presented for further improving these models in case when there are some in-domain unannotated data is available. Furthermore, implicitly annotated natural-language-like queries is used for testing the performance of the models, in a totally unsupervised fashion.

While in [156] a semantic parsing framework was proposed for question answering using a

knowledge base. A query graph that resembles sub-graphs of the knowledge base was defined which can be directly mapped to a logical form. This method leverages the knowledge base in an early stage to prune the search space and thus simplifies the semantic matching problem. Experimental evaluation showed that the proposed framework outperforms previous methods substantially, and has achieved an F1 measure of 52.5% on the WEBQUESTIONS dataset by applying an advanced entity linking system and a deep convolutional neural network model that matches questions and predicate sequences.

2.2.1.4 Other Parsers

Works such as [80], [127] and [136] proposed other parsing methods. In [80] authors proposed an algorithm of natural language text parsing for social network. The algorithm is used within a developed method of social network users' sentiment evaluation. Application of the proposed algorithm and technique was demonstrated on experimental data from Twitter social network. A special indicators were proposed and evaluated to estimate the accuracy of the algorithm in the experimental analysis stage in which the average value of the relative difference between total sentiment score obtained using the algorithm has manually reached 28.32%.

Furthermore, authors in [127] and [136] used different machine learning algorithms. In [127] proposed a recursive neural network framework to parse natural language and learning vector space representations. The proposed framework is based on context-sensitive recursive neural networks (CRNN) in which these networks can induce distributed feature representations for unseen phrases and provide syntactic information to accurately predict phrase structure trees. Furthermore, the representation of each phrase help in capturing semantic information. Results showed that the proposed framework has achieved F-measure of 92.1% on the Wall Street Journal dataset for sentences up to length 15. Finally, authors in [136] proposed a method for improving parser portability by combining parse re-ranking with data-defined kernels. This method is used to define a kernel over parse trees. Using SVM and a neural network probabilistic model as a classifiers; the performance has improved over the probabilistic model alone. In addition, this classifier is used to re-rank the top parses produced by the probabilistic model on the target domain. Experiments with a neural network statistical parser have demonstrated that this method helped in improving the parser accuracy on the target domain, without any significant increase in computational cost.

2.2.2 Tagging

In the following sub-sections, a detailed review on tagging is presented in addition to the different types of taggers.

2.2.2.1 General PoS Taggers

Many studies proposed taggers and tagging approaches; most of them have been developed for general PoS tagging. Works like [153], [100] introduced joint PoS tagging and dependency parsing. Authors in [153] introduced an approach that combined PoS tagging and dependency parsing using transition-based neural networks. In addition, to reduce the tagging, and labelling conflicts, three neural network based classifiers were designed. Experimental results showed that the proposed approach outperforms previous methods for joint PoS tagging and dependency parsing across a variety of natural languages.

Similarly, authors in [100] proposed a neural network based model that learns PoS tagging and graph-based dependency parsing jointly. The proposed model learns feature representations shared for both PoS tagging and dependency parsing tasks by using bidirectional Long Short-Term Memory (LSTM). The proposed model was tested on 19 languages from the Universal Dependencies project in which experiments showed that the proposed model outperforms the state-of-the-art neural network-based model for joint PoS tagging and transition-based dependency parsing.

Furthermore, authors in [138], [10] and [115] proposed a maximum-entropy-based PoS tagger. In [115] authors proposed a Maximum Entropy model. The proposed model trains from a corpus annotated with Part-of-Speech and uses many features to predict the PoS tag statistical model. The model has achieved an accuracy of 96.6%. While, in [138] authors proposed a maximum-entropy-based part-of-speech tagger. The proposed approach enrich the information sources used for tagging by incorporating into more linguistically features, such as features for the disambiguation of the tense forms of verbs and features for disambiguation particles from prepositions and adverbs. The results showed that the tagger achieved accuracy of 96.86% overall on the Penn Treebank.

In [10] authors Proposed a statistical part-of-speech tagger called Trigrams'n'Tags (TnT). Authors argued that a tagger based on Markov models performs at least as well as other current approaches, including the Maximum Entropy framework. Results showed that average part-

of-speech tagging accuracy is between 96% and 97%, depending on the language and the tag-set.

Works in [29] proposed a part-of-speech tagger based on Support Vector Machines Tool (SVMT). The proposed SVM-based tagger is robust and flexible for feature modelling, trains efficiently with almost no parameters to tune, and is able to tag thousands of words per second, which makes it suitable for real applications. Results showed that the SVM accuracy tagger significantly outperforms the TnT tagger, and has achieved an accuracy of 97.2% on the WSJ corpus, which is comparable to the best taggers reported up to date. In addition, in [30] SVMT was applied to a Spanish corpus exhibiting a similar performance with accuracy of 96.89%.

Other works like [137] proposed a part-of-speech tagger that demonstrates explicit use of both preceding and following tag contexts via a dependency network representation. Furthermore, the proposed tagger uses lexical features such as jointly conditioning on multiple consecutive words. The result of the experiments showed that the tagger achieved a 97.24% accuracy on the Penn Treebank WSJ.

Authors in [65] proposed part-of-speech taggers based on hidden Markov models, which adopt a less strict Markov assumption to consider rich contexts. In experiments, the Brown corpus were used which consists of 1,113,180 words and 53,885 sentences and is tagged with 82 PoS tags, which was segmented into two parts, the training set 90% and the test set 10% in a way that sentence in the test was extracted from every 10 sentence. Results showed that models with rich contexts achieved relatively high accuracy and some models assuming joint independence showed better results than the corresponding HMMs. In [131] authors proposed unsupervised part-of-speech (PoS) tagging by using an exact estimation method for learning anchor HMMs from unlabeled data.

In [67] authors presented a method for unsupervised part-of-speech tagging that considers a word type and it PoS tags as a primary element of the model. Results showed that the type-based tagger rivals state-of-the-art tag-level taggers which employ more sophisticated learning mechanisms to exploit similar constraints.

Authors in [61] proposed a neural framework that can infer meaningful word representations from the raw character stream. The proposed framework relies on two modelling stages which are a convolutional network and a prediction stage. The framework was evaluated on a PoS and morphological tagging task for German corpus. Experimental results showed that the convolutional network can infer meaningful word representations, while for the pre-

diction stage, a well-designed and structured strategy allows the model to outperform the state-of-the-art results, without any feature engineering. While, in [123] authors presented a new part-of-speech tagger for domain adaptation called FLORS. The proposed tagger input representation consists of three simple types of features which are, distributional count features and two types of binary features, suffix and shape features and uses SVM as a classifier. These representations work well for unknown words and for known words with unseen tags.

2.2.2.2 Domain Specific Taggers

Few taggers and tagging approaches have been developed for specific domains like web queries [54], [82], [27] and Twitter [31].

In [31] authors proposed a PoS tagger for Twitter to address the problem of part-of-speech tagging. A tag-set was developed using 1,827 tweets that were manually tagged. The set was randomly divided into a training set of 1,000 (14,542 tokens), a development set of 327 (4,770 tokens), and a test set of 500 (7,124 tokens). Results showed that the proposed tagger achieved 90% accuracy. Authors in [54] proposed a PoS tagging method for Web search queries using the sentence level morphological. Experimental results showed that the proposed method outperforms those using existing NLP tools and the state-of-the-art method. While, in [122] a new probabilistic tagging method was proposed called TreeTagger. The proposed tagging method uses decision tree to estimate the transition probabilities. This method has achieved 96.36% accuracy on Penn Treebank data.

Moreover, authors in [82] introduced two models for deep parsing of web search queries. The first model uses a grammar for generating multisets called a context-free multiset generating grammar (CFSG). While the second model consists of a parser was designed for parsing this type of grammar and a discriminative re-ranking module based on a support vector machine. Experiments showed that the first model outperforms a basic model, which is based on Conditional Random Fields when there is a small amount of training data. While the second hybrid model outperforms the other two modules regardless of the size of the training data. In [27] authors proposed a model to train a search-query PoS tagger from search-logs. The proposed model transfer the context from relevant snippet sets to query terms. Experiments showed that the model achieved more than 20% relative error reduction.

Finally, in [110] a tag-set was proposed that consists of twelve universal PoS categories. In addition, a mapping from 25 different tree-bank tag-sets to this universal set has been

developed. As a result, when combined with the original tree-bank data, this universal tag-set and mapping produced a dataset consisting of common PoS for 22 different languages. Two experiments have been conducted, to provide a language comparison, the same supervised PoS tagging model was trained on all of the treebanks and evaluated the tagging accuracy on the universal PoS tag-set. Second, universal PoS tags that were automatically projected from English have been used as the starting point for unsupervised grammar induction, producing completely unsupervised parsers for several languages.

To address the limitation of most parser and tagger methods which suffer from the problem of domain adaptation and do not take into consideration the syntax structure of the text. A domain-specific syntax parsing and tagging approach has been developed that uses not only generic PoS tags but also domain-specific PoS tags, grammatical rules, and domain knowledge. In addition, a tag-set that contains more than 10,000 words that could be used in different IR domains has been created. A detailed description is provided in chapter 3

2.3 Web Search Queries

Search engines are the most popular information retrieval applications. Despite that search engines try to improve the user experience and the technology used in finding relevant results, many difficulties are still faced because of the continuous increase in the amount of web content.

Semantic search has improved the information retrieval methods by looking at different perspectives, such as the meaning of words, yet search engines are still not capable of inferring the meaning of a term from the query it is contained in, which leads to ambiguity and retrieval of irrelevant information.

One major task in identifying the intent of a user's query is the classification of the query type. There are several taxonomies of web queries [2, 3, 9, 12, 53, 69, 99], of which Broder's taxonomy [12] is one of the most commonly used. It includes three main types: informational, navigational and transactional queries.

Different approaches and methods have been used to classify queries and to identify users' search intent by using: (a) the characteristics of each query type [12], [49], [150], [15], (b) users' behaviour by analyzing the query logs [119], [3], [9], [128] and (c) click through data [2], [69], [77].

In addition, machine learning algorithms have been used in the classification of different query types [47], [43], [154], [7], [86]. Furthermore, research such as [96], [121] and [4] analyzed the linguistic structure of web queries by applying techniques from natural language processing, such as part-of-speech tagging.

Web query search became more structurally complex over time [121], leading to the fact that two queries with overlapping sets of terms may reflect two totally different intents. To distinguish between these, users' behaviour or user clicks were used; however, these alone could be misleading in identifying the intent of a query [128].

In the following sub-sections, a detailed review of previous works on query classification is presented. In addition, the different proposed categories of query types and their characteristics are outlined in Section 2.3.1 and 2.3.2 respectively, while previous works on query classification methods are outlined in Section 2.3.3.

2.3.1 Queries Categories

Different categories of web queries according to user intent were defined, which are summarised in Table 2.1, and discussed below.

Web queries were classified by [99] by purpose, method, and content. The categories for the purpose of a query were defined as: (a) find, (b) compare or choose, and (c) understand. The methods were categories as: (a) explore, (b) monitor, (c) find, and (d) collect. The content referred to the topic of the query, e.g. education, news, for which ten categories were defined.

Broder's categories of web queries [12] are most commonly used in query classification. According to [12] web searches based on users' intent are classified into three categories: (a) Navigational, i.e. the intent is to reach a particular site, (b) Informational, i.e. the intent is to acquire information, and (c) Transactional, i.e. the intention is to perform a web-mediated activity, e.g. buy, download.

Broder's categories were extended by [119] and [49] by adding sub-categories. In [119] sub-categories were added for the informational and transactional categories, while [49] added sub-categories for all three types of queries. In [69], Broder's categories [12] were extended with two others, commercial and local.

Authors in [119] replaced the transactional queries with a category called resource queries, which they argue is broader than the transactional queries. The expansion of the taxonomy

TABLE 2.1 Summary of user intent categories for web queries

Authors	Categories of user intent
Morrison et al.,	Purpose: Find, Compare/Choose, Understand
2001 [99]	Method: Explore, Monitor, Find, Collect
	Content: Business, Education, News, etc.
Broder, 2002 [12]	Informational, Navigational and Transactional
Rose et al., 2004 [119]	Informational: Directed Closed, Directed Open, Undirected, Advice, Locate, List
	Navigational
	Transactional: Download, Entertainment, Interact, Obtain
Baeza-Yates et al., 2006 [3]	Goals: Informational, Not informational, Ambiguous
	Topics: Art, Games, Kids and Teens, Reference, Shopping, World, Business, Health, News, etc.
Kellar et al., 2006 [53]	Information Seeking: Fact Finding, Information Gathering, Browsing
	Information Exchange: Transactions, Communications
	Information Maintenance: Maintenance
Jansen et al., 2008 [49]	Informational: Directed (Closed or Open), Undirected, Find, List, and Advice
	Navigational: Navigation to Transactional, Navigation to Informational
	Transactional: Obtain (Online or Off-line), Download (Free or Not free), Results Page (Links or Others), Interact
Ashkan et al.,	Commercial
2009 [2]	Non-commercial: Navigational, Informational.
Calderon-Benavides et al.,	Genre: News, Business, Reference, Community
2010 [15]	Topic: Arts & Culture, Beauty & Style, Cars & Transportation, Computers & Internet, Education etc.
	Task: Informational, Not Informational, Both
	Objective: Resource, Action
	Specificity: Specific, Medium, Broad
	Scope: Yes, No
	Authority Sensitivity: Yes, No
	Spatial Sensitivity: Yes, No
	Time Sensitivity: Yes, No
Sushmita et al.,	Domain: Image, Video, Map
2010 [134]	Genre: News, Blogs, Wikipedia
Lewandowski et al., 2012 [69]	Informational, Navigational, Transactions, Commercial, Local
Bhatia et al., 2012 [9]	Ambiguous, Unambiguous but underspecified, Information gathering, Miscellaneous.

in [49], however, reverted the name to transactional, while keeping the sub-categories initially proposed by [119] under the name of resource queries.

In [3], user goals and categories of topics were used for query classification. The user goals were divided in three categories: (a) informational, (b) not informational, and (c) ambiguous. For topics, 18 categories were used.

Web information tasks were classified by [53] according to three types of information goals: (a) information seeking, (b) information exchange, and (c) information maintenance. Each of these goal categories contains information tasks.

In [2], the focus was on identifying if the user had the intention to purchase or utilise a commercial service. From this point of view, two categories were defined: (a) commercial and (b) non-commercial. The second category was further split into two sub-categories from Broder's classification [12], i.e. navigational and informational.

In [15] several dimensions on user intent were defined based on the argumentation that a user's intent is complex and that the complexity is considerably reduced when looking at smaller, better defined aspects. By combining this classification with Broder's one [12] and the one by [134] (see below) another multi-dimensional classification was proposed by [145].

A classification according to the types of documents sought by a user was proposed in [134], by using the domain (image/video/map) and genre (news/blogs/wikipedia). With a focus on results diversification, [9] proposed four types of queries: (a) ambiguous, (b) unambiguous but underspecified, (c) information gathering, and (d) miscellaneous. The different categories of user intent reflect different perspectives on ways to improve query classification.

In the next sub-section we focus mainly on query classification using Broder's categories [12] or their variations [119], [49], as this is the most popular user intent taxonomy and the proposed framework is validated using these intent categories. Previous works related to query classification based on Broder's categories [12] and using machine learning approaches, are summarised in Table 2.2.

2.3.1.1 Broder's Query Classification

Web queries are classified according to their intent into three categories informational, navigational and transactional (Broder, 2002). Some queries can belong to more than one of these categories others can belong to neither. These categories could be defined as follow:

TABLE 2.2 Research using Broder's categories of web queries

Authors	Inf.	Nav.	Trans.
Rose, et al., 2004	X	X	X
Lee, et al., 2005	X	X	
Liu, et al., 2006	X		X
Baeza-Yates, et al., 2006	X		
Jansen, et al., 2008	X	X	X
Mendoza, et al., 2009	X	X	X
Kathuria, et al., 2010	X	X	X
González-Caro and Baeza-Yates, 2011	X		
Hernandez, et al., 2012	X	X	X
Lewandowski, et al., 2012	X	X	X
Figueroa, 2015	X	X	X

- Navigational Queries: queries in this category have one right result since the purpose of such query is to reach a particular site, for example *"British airways homepage"*. Furthermore, in this type of queries the user usually has a certain website in mind but either does not know the URL or may think that a particular website exists.

- Informational Queries: the purpose of this type of query is to find information, learn how to do something or just answer a question. In addition, this information is available on the web in a static form and no further interaction is needed. Furthermore, topics of these type of queries are usually broad and general such as *"Las Vegas"*, others are specific like *"Brain Cancer"*, usually there is no particular web page containing all the information needed; users have to acquire the information needed from multiple web pages.

- Transactional Queries: the purpose of this type of query is to find a site and further interaction may be required like downloading a software or buying a certain product online, also the purpose may be to acquire something not to find information about it but to print it out or just to look at it on the screen such as *"Music lyrics"* or *"Recipes"*.

2.3.1.2 Queries Extended Classification

Informational, navigational and transactional queries could be classified into three level of hierarchical taxonomy, shown in Figure 2.1, with level one consider the top level including informational, navigational and transactional [12]. Each of these types have multiple level

two classifications and some can also have a third level classifications [49], this extended classification of level two and three are similar to [119].

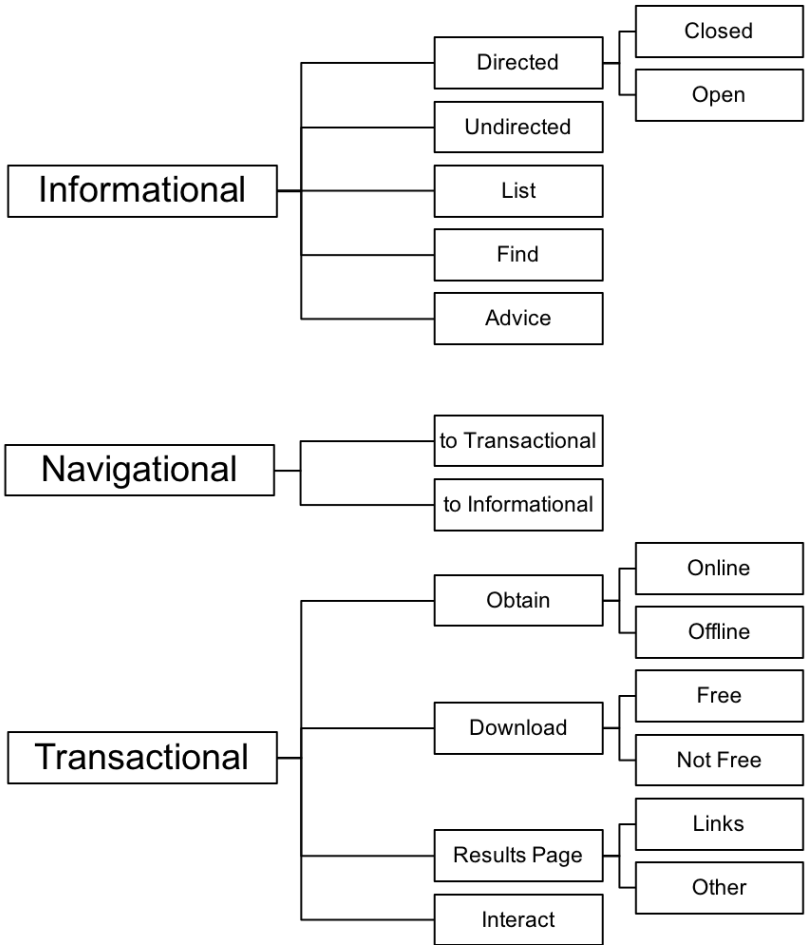

FIGURE 2.1 Web Queries Classification

Works in [119] extended informational query in Broder's classification [12] by adding five sub-categories: Directed-open and Directed-closed, Undirected, List, Advice, and Locate. The Resource category is also extended to contain four sub-categories: Download, Entertainment, Interact and obtain.

According to [49] transactional query is further classified to: Download, Interact, Obtain and Results Page. These categories have level three sub-categories: Download-Free, Download-Not Free, Obtain-Online, Obtain-Offline, Results Page-Links and Results Page-Other. In addition, navigational query has level two sub-categories: Navigational-to-Transactional and Navigational-to-Informational. Moreover, authors in [49] have provided characteristics for informational, navigational and transactional category that helped to define the queries in each category.

2.3.1.3 Informational Query

Informational query has five sub-categories: Directed, Undirected, List, Advice, Locate and Find.

1) Informational-Directed: the purpose of this category is to answer a specific question, both open and closed ended or to learn something in particular about a certain topic. This category has level two sub-categories:

    a) Informational-Directed-Open: the goal of this category is to find information about two or more topics, it may take many forms either a question to get an answer for an open-ended question or one with unconstrained depth. Examples: *"Why recycling is important?"* and *"Ants communication"*.

    b) Informational-Directed-Closed: in this category queries could be a question to find information about one specific topic or to find one specific or unambiguous answer. Examples: *"Capital of Italy"* and *"What is a real number?"*.

2) Informational-Undirected: most queries in this type are related to science, medicine, history and news and celebrities, the goal of this category is to know anything and everything about a topic. [119]. Examples: *"Michael Phelps"*, *"Civil War"* and *"Hydrofluoric Acid"*.

3) Informational-List: the goal of this type of queries is to find a list of suggested websites or candidates or list of suggestions for further research, also plural query terms are a highly reliable indicator of this category [119]. Examples: *"list of animated movies"*, *"Wales universities"* and *"things to do in London"*.

4) Informational-Find: the objective of this category is to locate or find something in the real world like a service or a product. Most shopping or product queries have the locate goal [119], for example: *"Apple store location in London"* and *"Cheap Samsung Mobiles"*.

5) Informational-Advice: the purpose of this category is to get advice, suggestions, ideas or instructions about something and may take many forms like a question. Examples: *"How to quit smoking"* and *"Writing a story"*.

2.3.1.4 Navigational Query

Navigational query has two sub-categories: Navigational-to-Transactional and Navigational-to-Informational.

1) Navigational-to-Transactional: in this type of query the user is searching for transactional web page or the URL is for a transactional web page. Examples, *"ebay.com"* and *"amazon.com"*.

2) Navigational-to-Informational: in this type of query the user is searching for informational web page or the URL is for a informational web page. Examples, *"yahoo.com"* and *"google.com"*.

2.3.1.5 Transactional Query

Transactional query has the following four sub-categories: Obtain, Download, Interact, and Results Page.

1) Transactional-Obtain: the objective of this type of queries is to obtain specific resource or object, not to learn some information but just to use the resource itself. This category has the following level two sub-categories:

    a) Transactional-Obtain-Online: in this type of queries the user might search for something to just look at it on the screen, meaning that the resources will be obtained online. Examples, *"Cupcakes Recipes"* and *"Sam Smith songs lyrics"*.

    b) Transactional-Obtain-Offline: in this type of queries the user might search for something to print or save to use it later offline, meaning the resources of this type of queries will be obtained offline and may require additional action by the user. Examples: *"Flowers Wallpapers"* and *Windows 10 screen-savers"*.

2) Transactional - Download: the resource of this type of query is something that needs to be installed on a computer or other electronic device to be useful like finding a file to download. This category has level two sub-categories:

    a) Transactional-Download-Free: the download-able file is free. Examples: *"Free image editor downloads"* and *"Free online games"*.

b) Transactional-Download-Not Free: the download-able file is not necessarily free. Examples: *"The time keeper book download"* and *"Celine Dion songs download"*.

3) Transactional-Interact: this type of queries occur when the intended result of the search is a dynamic web service, and requires further interaction with a program or a resource. Examples: *"Currency Converter"*, *"Buy mobile phones"* and *"Weather"*.

4) Transactional-Results Page: the objective of this category is to obtain resources that can be saved, printed, or read from the search engine results page. This category has level two sub-categories:

   a) Transactional-Results Page-Links: the resources of this kind of queries appear in the URL, title or summary of the search engine results page. For example: *"Searching for a title of a conference paper to locate the page numbers"*.

   b) Transactional-Results Page-Other: the resources of this kind of queries do not appear on the search engine results page but somewhere else on the search engine results page. For example: *"Spelling check of a certain term"*.

2.3.2 Characteristics of Web Search Queries

Informational, navigational and transaction query; each has it own characteristics; queries in each type differ from one another, according to [49] the identification of the characteristics of each query type will lead to real world classification.

2.3.2.1 Informational Search Characteristics

One of the major characteristics of informational query is the use of natural language phrases [49]. In addition, queries for such search may consist of informational terms like *"List"* and *"Play-list"* and searches related to Advice, help and guidelines like *"FAQs"* or *"How to"*. Furthermore, queries may contain question words like *"Who"*, *"What"*, *"When"*. In addition, queries may consist of words related to ideas and suggestions terms, recent information and news, topics related to science, medicine, history and celebrities [119]. Moreover, some queries consisting of multimedia like videos are considered informational like *"How-to-do"* videos.

2.3.2.2 Navigational Search Characteristics

Navigational queries contain, organization, business, company and universities names, domain suffixes like *".com"*, *".org"* and domain prefixes such as *"www"* or *"http"* and *"web"* as the source. In addition, some navigational queries may contain URLs or parts of URLs [49].

Furthermore, most queries consisting of people names, including celebrities, are not considered navigational. According to [119] the goal or objective of searching for a celebrity is usually not just visiting a specific site and a search for a celebrity such as *"Merly Streep"* will result in a fan or media sites.

2.3.2.3 Transactional Search Characteristics

According to [49] queries in transactional search are related to obtaining terms like *"lyrics"*, *"recipes"* and *"patterns"*, download terms such as *"software"*. In addition, transactional queries might contain *"audio"*, *"video"* and *"images"*.

2.3.3 Query Classification Methods

In the following sub-sections, a detailed review on query classification is presented and the different types of methods used for the identification and classification of users' search intent.

2.3.3.1 Features-Based Methods

Different approaches and methods have been used to classify queries and to identify users' search intent.

A survey of AltaVista users was used in [12] as a method to classify user's query manually in order to determine the type of queries. The survey was conducted online and users were selected randomly, the data consisted of 3,190 valid results and achieved a response ratio of about 10%. In this survey users were asked to describe the purpose of their search and in order to distinguish between navigational and non-navigational queries questions such as, "why they conducted this search?" and "what they are looking for?" were asked. The result showed that 24.5% of queries were navigational and 68.4% were non-navigational.

In addition, queries that were neither transactional nor navigational were assumed to be informational since the authors could not distinguish between informational and transactional queries using a simple question. The final result of the survey showed that 24.5% of queries were navigational, 39% were informational queries and 36% were transactional queries. Furthermore, a random set of 1,000 queries were analyzed from the AltaVista daily log of user queries. Result of the analysis showed that 20% of queries were navigational, 48% were informational and 30% were transactional.

In [49] authors proposed a comprehensive classification of user intent for web searching. The classification consists of three hierarchical levels of informational, navigational, and transactional intent. Furthermore, a software was developed that automatically classified queries using web search engine log of over a million and a half queries.

Results showed that more than 80% of web queries were informational and about 10% of the queries were navigational and transactional. In addition, 400 queries from Dogpile transaction log (dogpile.com) were randomly selected and manually coded to validate the proposed approach. Results showed that 74% of the queries were successfully classified and the remaining 25% were vague or multi-faceted queries, which highlighted the need for probabilistic classification.

Works in [15] analyzed and characterized a wide range of facets and dimensions in order to identify user's search intent. These dimensions/facets are: genre, objective, specificity, scope, topic, task, authority sensitivity, spatial sensitivity and time sensitivity. In addition, a sample of $5,249$ queries from the TodoCL (todocl.cl) search engine query log was used. These sample was manually classified by a group of judges and in order to estimate the reliability of the set of assessments, two judges classified 10% of the queries. The analysis of the manual classification of queries showed that dimensions such as scope, topic and objective are easier to determine than genre and task.

Furthermore, some works used users' behaviour by analyzing the query logs to classify queries. Authors in [119] argued that user goals can be deduced from looking at user behaviour available to the search engine like the query itself and result clicked. Based on that, a tool was created that provides this type of information. In addition, three sets of approximately 500 queries were randomly selected from AltaVista query logs and analyzed. The limitation of this approach is that the goal-inferred from the query may not be the user actual goal.

Manual classification is subject to some errors, therefore authors in [3] created a software

tool to help evaluating the result of the automatic classification, the software allows the user to select the goal and category. Moreover, a manually classified data was used to evaluate the results of the automatic classification. A log sample of 6,042 queries was used from the Chilean web search engine TodoCL2 (todocl.cl) and the test set was built based on a team of people who performed a manual classification of the queries. The manual classification of the queries was made in order to have a reference point and then supervised and unsupervised learning techniques were applied. Results obtained showed that combining supervised and unsupervised learning is a good alternative to find user's goals.

Authors in [128] proposed a query enrichment method by mining the documents clicked by users and the relevant follow up queries in a session. In addition, documents and the queries were mapped using a text classifier into predefined categories and extracting features from the processed data. Moreover, Support Vector Machine (SVM) algorithm were used for the classification process. Experimental results showed that when combining the two sets of features, the proposed approach achieved effectiveness of 86% in terms of accuracy and significantly improved the click-based method by 5.6% and the session-based method by 4.6%.

Moreover, click-through data has been used for the identification and classification of different queries and users' search intent. In [2] authors classified 1,700 queries and manually labelled the selected queries then used ads click-through and query features to determine the query intent. A methodology to use the combination of ads click-through and query features with the content of search result page was developed in order to determine the intention underlying queries, especially commercial intent. Result showed that ad click-through features, query features, and the content of search engine result pages are together effective in detecting query intent. The ad click-through features improved the accuracy of detecting different query intents. Result showed that, 42% of the queries were labelled as commercial and 58% were labelled as non-commercial, while 60% of the queries were labelled as navigational and 40% were labelled as informational.

Authors in [69] analyzed click-through data to determine Commercial and Navigational queries. In addition, crowd-sourcing approach was used to classify search queries. First, using approximately 50,000 queries; a large-scale classification study was conducted. Then, a click-through data was used from a search engine log to validate the judgments given by the jurors from the crowd-sourcing study. Finally, an online survey was conducted on a commercial

search engine's portal.

Furthermore, the crowd-sourcing approach using jurors who classified queries originating from other users and the questionnaire approach using searchers who were asked about their own query that they just entered into a web search engine, lead to unsatisfying results. In addition, results obtained from the user survey showed that users have difficulty understanding query classification tasks and a clear recommendation on which approach to use when classifying query intents could not be given. The Final results showed that the automatic approach performed well on navigational queries, and to some degree on commercial queries. The crowd-sourcing approach and the online survey lead to mixed results which indicates that when using one of these approaches, reliability checks should be applied to avoid misclassified queries.

Works in [77] used click-through data to identify users' goals behind web search queries. Based on user logs, which contain over 80 million queries and corresponding click-through data, two novel features extracted were proposed from click-through data and a decision tree based classification algorithm for identifying user queries. The experimental evaluation showed that the algorithm could correctly identify the goals for about 80% of web search queries. In addition, the reliability and scalability of the classification method were verified by obtaining part of query logs from a widely used Chinese search engine Sogou (sogou.com).

In order to verify the effectiveness of the identification algorithm, a test set has been developed. The test set is composed of 81 informational/transactional queries and 152 navigational queries. Moreover, to judge the effectiveness of the query type identification task precision/recall was used. Precision and recall values are calculated separately for the two kinds of queries, and then F-measure value was combined to judge the overall performance.

The query analysis by [66] was done by using two types of features: past user click behavior and Anchor-link distribution. Authors proposed two types of features, past user-click behavior and anchor-link distribution. Results showed that the combination of these two techniques could correctly identify the goals for 90% of the queries. One limitation of this study is that the experiment was conducted on a potentially biased dataset.

2.3.3.2 Features and Machine Learning-Based Methods

Authors in [52], [9] and [25] used a variety of query features to automatically classify the user intent behind web queries in addition to machine learning algorithms.

Authors in [52] automatically classified different users' intent using a k-means clustering approach based on a variety of query traits. The results showed that more than 75% of the web queries which were clustered into eight classifications are informational and about 12% each for navigational and transactional. In addition, results showed that web queries fall into eight clusters, six primarily informational, and one each of primarily transactional and navigational.

Works in [9] presented an analysis of a commercial web search engine log. Queries were analyzed based on their click entropy and popularity. In addition, a query taxonomy was proposed based on their diversification requirements. Web search queries were automatically classified into one of the classes of the proposed taxonomy (Ambiguous, Unambiguous but Underspecified, Information Browsing and Miscellaneous).

Furthermore, a various query-based, click-based and reformulation-based features were utilized for the query classification task and achieved strong classification results. The utilized features that were described from the users' input query, click-through information and query reformulations have achieved an overall precision of 74.8% and recall of 73.3% for the automatic query classification task.

In [25] authors used assorted features for automatically detecting the user intent behind web queries. Results showed that linguistically motivated features such as WordNet semantic relations and specialized models like NERQ or using caseless models as a fallback alternative helped in improving the recognition of the intent behind search queries.

Other works used machine learning algorithms for the classification of different query types. In [7] authors proposed a framework for automatic web query classification that combines a small seed manual classification with techniques from machine learning and computational linguistics. Furthermore, three approaches were examined for the categorization of the general web queries; matching against a list of manually labelled queries, supervised learning of classifiers, and mining of selection preference rules from large unlabelled query logs. The combined method accurately classified 46% of the queries outperforming the recall of the single approach by nearly 20%, with a 7% improvement in overall effectiveness. A validation set of 5,283 queries were randomly sampled and used from the query stream and was manually classified by a team of editors at AOL.

Moreover, authors in [86] proposed three vector representations for queries based on click-through data and descriptive text. In addition, four relevant factors were identified which are; frequency of terms, (TF), inverse frequency in documents (Idf), user preferences (Pop), and

reading time of selected documents (Time). The performance of the three representations over a set of queries categorized by experts were evaluated using SVM. Furthermore, a set of 2,000 queries was manually classified by a team of expert using the categories proposed by [12]. As a result of the classification process, 1,953 queries were labelled by consensus.

The results showed that 52% of the queries were informational, 33% navigational and 15% transactional. In addition, 70% (1,367 queries) of the manually classified queries were considered as training data, leaving the remaining 30% for evaluation (586 queries). Experimental result showed the proposed classifiers can effectively identify the intent of past queries with high precision. In addition, the third method achieves good results considering error rates as the performance measure.

Works in [43] proposed a solution that automatically classifies queries using the text included in the query, based on the features and characteristics described by [12], [49] and [150]. In addition, a set of features extracted from the terms included in the query was used, without any external or additional information. The features proposed were automatically extracted from two different corpora then machine learning algorithms were implemented to validate the accuracy of the classification and to evaluate the results.

Two query datasets were used from Million Query Track of TREC; MQ2007 (with 1,692 queries) and MQ2008 (with 784 queries). Results showed that informational queries account for 82%, navigational 11.5% and transactional 6.5% for the MQ2007 dataset. While, informational queries accounted for 82%, navigational 11% and transactional 7% for the MQ2008 dataset. Precision, Recall and F-measure was used to evaluate the performance of the algorithms for each user intent category. SVM obtained better results on the informational category and Navie Bayes is better for navigational and transactional categories.

Works in [47] presented a data-driven methodology to disambiguate a query by suggesting relevant sub-categories within a specific domain by finding correlations between the user's search history and the context of the current search keyword. Neural Networks and Naive Bayes classifier is applied to learn the category of a given query from a training set.

Authors in [154] stated that the fine-grained topics in the same category of the taxonomy may be textually more relevant to the topics in other categories, this phenomenon may affect the performances of most traditional classification methods. They presented K-Nearest topic classifier to enhance the performances of traditional query classifiers, by detecting millions of fine-grained query topics from two years of click-logs then calculating the K most relevant

topics and select the label by majority voting, then try to use this label to improve the results of classical query classification methods.

2.3.3.3 Linguistic Structure-Based Methods

Web queries and user's search intent has been identified and classified by analyzing the linguistic structure of web queries. Authors in [71] stated that the semantic intent of web queries not only involves identifying their semantic class but also understanding their semantic structure. Accordingly, their research involved the analysis of the semantic structure of noun phrase queries.

While, [121] examined the structure of web queries by applying techniques from natural language understanding; this analysis showed that queries have distinct properties of their own and are not just some form of text between random sequences of words and natural language.

Furthermore, according to [4] queries exhibit their own partially unique linguistic structure; their analysis of queries was based on the syntax of part-of-speech tag sequences. Their results showed that query part-of-speech tagging can be used to create significant features for improving the relevance of web search results and may assist with query reformulation.

Authors in [96] introduced a new solution to automatically identify and classify the user's queries intent by using Search Type Patterns. The proposed approach takes into consideration query structure along with query terms. Experiments showed that this approach achieved classification accuracy of 85.5%.

Unlike the previous approaches, a formal grammar-based framework was proposed for query classification (GQC), which exploits the structure within the text through a new representation using general and domain-specific syntactic categories. Details of the framework and its use on query classification are given in chapter 4.

2.4 Question classification

Question-answering has become one of the most popular information retrieval applications. Questions Classification (QC) plays an important role in question-answering systems and one of the major tasks in the enhancement of the classification process is the identification of questions types.

Despite that most Question-Answering Systems (QASs) try to improve the technology

used in retrieving relevant results, many difficulties are still faced because of the continuous increase in the amount of web content and the low response rate to many questions [78], [79]. The goal of the question classification process is to accurately assign labels to questions based on an expected answer type [87].

The task of generating answers to the users' questions is directly related to the type of questions asked [98]. Hence, the classification of the questions performed in QASs directly affects the answers. Results show that most errors happen due to miss-classification of questions performed in QASs [98]. Authors in [13] performed function oriented classification of questions by integrating pattern matching and machine learning techniques, while [8] classify questions by taking account of their expected types of responses. In addition, [58] stated that question type is defined as a certain semantic category of questions characterized by some common properties.

Recent studies classified users' questions using different features like bag-of-words [159], [72], [155], [88], semantic and syntactic features [155], [41], [129], and uni-gram and word shape features [48]. Authors in [48] stated that features are the key to obtain an accurate question classifier. Furthermore, in order to distinguish between different types of questions, many previous studies classified questions using different machine learning algorithms.

Support Vector Machine (SVM) is one of the most used algorithms [87], [14], [48], [40], [144], [42], [151]. According to authors in [88] combining an SVM classifier with semantic, syntactic and lexical features improves the classification accuracy. Other works like [159], [129], [88] and [97] used SVM in addition to other machine learning algorithms such as Naive Bayes, Nearest Neighbors and Decision Tree. Moreover, works like [120] and [141] used Neural Networks as the machine learning algorithm.

In the following sub-sections, a detailed review of previous works on question classification is provided. In addition, the different proposed question categories are outlined in Section 2.4.1, while previous works on question classification methods are outlined in Section 2.4.2

2.4.1 Questions Categories

Different categories of questions were defined, which are summarised in Table 2.3. According to authors in [58] the major question types are: factoids, list, definition, hypothetical,

causal, relationship, procedural, and confirmation questions. A factoid question is a question which usually starts with a Wh-interrogated word (What, When, Where, Who) and requires as an answer a fact expressed in the text body. On the other hand, a list question is a question, which requires as an answer a list of entities or facts; a list question usually starts as: List/Name [me] [all/at least NUMBER/some]. Furthermore, a definition question is a question, which requires finding the definition of the term in the question and normally starts with "What is". Related to the latter is the descriptive question, which asks for definitional information or for the description of an event, and the opinion question whose focus is the opinion about an entity or an event. A hypothetical question is a question, which requires information about a hypothetical event and has the form of "What would happen if". In addition, a causal question is a question which requires explanation of an event or artifact, typically starting with "Why". A relationship question asks about a relation between two entities, while a procedural question is a question which requires as an answer a list of instructions for accomplishing the task mentioned in the question. Finally, a confirmation question is a question, which requires a Yes or No as an answer to an event expressed in the question.

TABLE 2.3 Summary of user intent categories for questions

Authors	Categories
[58]	factoids, list, definition, hypothetical, causal, relationship, procedural, and confirmation questions
[13]	Fact, List, Reason, Solution, Definition and Navigation.
[14]	Advantage/Disadvantage, Cause and Effect, Comparison, Definition, Example, Explanation, Identification, List, Opinion, Rationale and Significance.
[73]	Abbreviation, Description, Entity, Human, Location and Numeric as coarse classes; and Expression, Manner, Color, City.

The classification in [13] was motivated by related work on user goal classification by Broder [12] and Rose and Levinson [119]. The proposed function-based question classification categories were tailored to general QA, containing six types, namely: Fact, List, Reason, Solution, Definition and Navigation. For the Fact type of question the expected answer will be a short phrase; these questions are asked to get a general fact as an answer. For the List type of

question each answer will be a single phrase or a phrase with explanations or comments; these questions are asked to get a list of answers. Furthermore, a good answer summary should contain a variety of opinions or comprehensive explanations for Reason type of question in which sentence-level summarization can be employed; these questions are asked to get opinions or explanations as the answer. For the Solution type of questions, the sentences in an answer usually have a logical order, thus the summary task cannot be performed on sentence level; these questions are asked to solve a problem. The Definition type of questions are asked to get a description of concepts as an answer; usually this information can be found in Wikipedia. If the answer is too long, it should summarized into a shorter one. Finally, Navigation type of questions are asked to find websites or resources; sometimes the websites are given by name and the resources are given directly.

Authors in [14] classified open-ended questions to 11 categories, which are: Advantage/Disadvantage, Cause and Effect, Comparison, Definition, Example, Explanation, Identification, List, Opinion, Rationale, and Significance. Advantages and disadvantages are questions that may require certain number, while Cause and Effect are questions that explain the effect of something on something else. Moreover, a Comparison question answer outlines differences and/or similarities between two or more entities. Furthermore, a Definition question requires a relatively short explanation or description (just few lines or few sentences). On the other hand, an Example question requires an answer that provides an example. An Explanation question provides more explanation or more details than the 'what' questions. Identification questions provide answers allowing the identification of something. The List question provides a list of points which may or may not be in sequence. Opinion questions give as answers personal opinions on a particular point or a statement supporting an argument or advocating against it. Finally, the answer to a Rationale question explains why a statement/question is true or false, while an answer to a Significance question explains the importance of something or why it may be important.

Many researchers focused on a particular type of question. For example, work in [45] focused on the "causal" question type, while works in [8, 87, 144] focused on factoid questions. Furthermore, most works are based on Li and Roth [73] classification of question [48, 64, 70, 72, 84, 87, 88, 101, 102, 144, 151, 159] in which these works focused on factoid questions since the categorization proposed by Li and Roth mainly deals with this type of question. Their two-layer taxonomy consists of a set of six coarse-grained categories which

are Abbreviation, Entity, Description, Human, Location and Numeric value, and fifty fine-grained ones, e.g., Abbreviation, Description, Entity, Human, Location and Numeric as coarse classes, and Expression, Manner, Color and City as fine-grained classes. This classification has a limitation since it deals with factoid questions only which is a very limited class of real world questions.

2.4.2 Question Classification Methods

Many recent studies classified users' question using different features like bag-of-words [159], [72], [155], [88], semantic and syntactic features [155], and uni-gram and word shape features [48]. Furthermore, to distinguish between different types of questions, many previous studies classified questions using different machine learning algorithms.

Authors in [48] proposed head word features, which is one single word specifying the object that the question seeks, and used two approaches to augment the semantic features of such head words using WordNet. In addition, other standard features were augmented, which means some features were increased, such as wh-word, unigram feature, and word shape feature.

In [155] a framework has been proposed, which integrates a question classifier with a simple document/passage retriever, and proposed context-ranking models. This method provides flexible features to learners (machine learning algorithms), such as word forms, syntactic features, and semantic word features. In addition, the proposed context-ranking model, which is based on the sequential labelling of tasks, combines rich features like full parsers, predefined syntactic patterns, and more training materials to predict whether the input passage is relevant to the question type.

The work in [72] used machine learning approaches, namely, different classifiers and multiple classifier combination methods by using compositive statistic and rule classifiers, and by introducing a dependency structure from Minipar and linguistic knowledge from Wordnet into question representation. In addition, features like the dependency structure, WordNet synsets, bag-of-words, and bi-gram were used. Also, a number of kernel functions were used and the influence of different ways of combining classifiers, such as Voting, AdaBoost, Artificial Neural Networks (ANN) and Transition-Based Learning (TBL), on the precision of question classification was analyzed.

In [39] a hybrid approach was proposed, named ATICM which is based on dependency tree analysis for automated answer type identification and classification by utilizing both syntactic and semantic analysis. This method contains a compact WordNet-based hypernym expansion strategy to classify identified question target words into question target categories. Result showed that ATICM approach has achieved an accuracy of 93.9% on the UIUC dataset and 92.8% on the TREC10 dataset.

In addition, authors in [144] proposed a method of using a feature selection algorithm to determine appropriate features corresponding to different question types. Moreover, they designed a new type of feature, which is based on question patterns; then applied a feature selection algorithm to determine the most appropriate feature set for each type of questions. The proposed approach was tested on the benchmark dataset TREC, using SVM for the classification algorithm.

In [87] a statistical classifier has been proposed which is based on SVM and uses prior knowledge about correlations between question words and types in order to learn question word specific classifiers, i.e. a what question will be classified with SVM$_{what}$. In addition, any data set, question ontology, or set of features can be used with this statistical framework.

Furthermore, [151] proposed a SVM-based approach for question classification. In addition, a dependency relations and high-frequency words are incorporated into the baseline system. Experiments on the UIUC corpus showed that the introduced features can improve the baseline system significantly in which the combination of top word and dependency relation features improved the accuracy to 93.4%.

Other works like [159] and [88] used SVM in addition to other machine learning algorithms. [88] proposed an approach for question classification through using three different classifiers, k-Nearest Neighbor (KNN), Naïve Bayes (NB), and SVM, using two kinds of features: bag-of-words and bag-of-ngrams. In order to train the learning algorithm, a set of lexical, syntactic, and semantic features were used, among which are the question headword, which is a word in a given question that represents the information that is being sought, and hypernym which is a word with higher level semantic concepts. Similarly, in [159] five machine learning algorithms were used, KNN, NB, Decision Tree (DT), Sparse Network of Winnows (SNoW), and SVM, using two kinds of features: bag-of-words and bag-of-ngrams.

SVM were also used in [14] for the classification of open-ended questions. They have stated that SVM could be trained to recognize the occurrence of certain keywords or phrases

in a question class and then, based on the recurrence of these same keywords, be able to correctly identify a question as belonging to that class.

Another classification approach has been proposed in [36] using SVM. According to the authors in this work an enormous amount of time is required to create a rich collection of patterns and keywords for a good coverage of questions in an open-domain application, so they have used support vector machines for question classification. The goal is to replace the regular expression based classifier with a classifier that learns from a set of labelled questions and represented the questions as frequency weighted vectors of salient terms.

Moreover, works like [120] and [141] used Neural Networks as the machine learning algorithm. [120] proposed a neural network for a question answering system. The proposed network is composed of three layers and one network: Sentence Layer, Knowledge Layer, Deep Case Layer and Dictionary Network. The input sentences are divided into knowledge units and stored in the Knowledge Layer.

In [74] a classification method was proposed for community question answering (CQA) system based on ensemble learning, using supervised learning and semi-supervised learning of different feature extraction methods like lexical semantic extension and different classifiers in the question classification, the supervised learning and the semi-supervised learning adopt three different classifiers, which are J48graft, J48 and Naïve Bayes. The experiments verified that the semi-supervised classification algorithm based on ensemble can effectively utilize a mass of unlabelled question samples to enhance the classification accuracy.

Finally, the proposed approach in [141] formulates the task as two machine learning problems, which are, detecting the entities in the question, and classifying the question as one of the relation types in the knowledge base. Based on this assumption of the structure, this approach trained two recurrent neural networks and outperformed state-of-the-art approaches by significant margins; the relative improvement reached 16% for web questions, and surpassed 38% for simple questions.

Unlike the previous approaches, a grammar-based framework was proposed for questions categorization and classification (GQCC) which deals with different types of questions and different domain categories by exploiting the structure of the question through using general and domain-specific grammatical categories and rules. Moreover, the grammar provides a flexible and powerful platform for integrating prior-domain information about each question category into the tagging and classification phases. The proposed framework is introduced in

2.5 Summary of Chapter

In this chapter, a literature review of related works of text classification methods and techniques was presented. Previous text classification methods are based on features such as bag-of-words (BoW) model and n-grams in addition to different machine learning algorithms such as SVM and NB. In addition, a full description of parsing and tagging approaches was provided. Different parsers have been developed in which the majority are grammar-based and dependency-based parsers while few parsers are semantic-based. Moreover, most tagging approaches are general PoS tagger while others have been developed for specific domains such as twitter.

Furthermore, a detailed overview of queries classification methods and techniques was highlighted in which query classification using Broder's categories is the most popular user intent taxonomy. In addition, query features such as users' logs and click through data are the most used methods for identifying and classifying different types of query. Finally, a detailed overview of questions classification methods and techniques was highlighted. Most works focused on the identification and classification of factoid type questions in which the majority are based on Li and Roth classification of question. In addition, features like bag-of-words, n-grams, semantic and syntactic features are mostly used for question classification.

CHAPTER 3

A Customizable Grammar-Based Framework For User-Intent Text Classification

This chapter introduces the Customizable Grammar Framework (CGF) for user-intent text classification. The chapter is organised as follows. First, an overview of the framework is presented in Section 3.1 where the three main phases of CGF are defined: (1) grammar; (2) parsing and tagging; (3) learning and classification. Following that, a detailed description of the CGF is presented in Section 3.2 and Section 3.3 summarizes the chapter.

3.1 Overview

A Customizable Grammar Framework (CGF) is proposed to address the limitations of general approaches in text classification and incorporate domain-related information without increasing the complexity of the textual representation and computation, as well as take into account the structure of the text. The general framework is described below, while its use for the query and question classification problem is detailed in the following chapters.

CGF combines domain knowledge with a formal grammar by the use of grammatical rules and patterns. Unlike typical bag-of-words text representations, CGF takes into consideration the grammatical structure of the text. The aim of this approach is to create a general framework that could easily be modified and applied to different domains by creating a specific formal grammar for each.

The CGF framework introduces a new representation for textual data that aims to preserve

the grammatical structure of the text and makes use of a formal grammar to transform the text into this new form of representation, as outlined below:

- Each word/term is represented as its syntactic category;
- The text is represented as an ordered series of syntactic categories, which we call syntactic patterns;
- A formal grammar is defined to transform the text into this representation;
- The formal grammar contains in addition to typical syntactic categories of English grammar, domain-related syntactic categories.

This representations is different from the typical bag-of-words approaches, where all the words of all instances (e.g. documents, queries) become the features and the values of the features are metrics of term frequency, of which the most popular is $tf - idf$ (term frequency-–inverse document frequency). PoS-tagging features, i.e. the syntactic categories of words, can also be used to represent text, either on their own or in combination with the bag-of-words features. The representation, however, is the same, i.e the features are the PoS-tags and the values of the features are metrics of term frequency. This representation does not preserve the order of the words in the original instances and leads to large and sparse datasets. For the later reason, features with low frequencies are typically removed, risking the removal of relevant information.

The proposed representation addressed the limitations of the bag-of-words approach by preserving the order of the words and by representing an instance as a syntactic pattern, in which the maximum length of an instance is the number of words in that instance, although that number may be even lower as some groups of words are treated as expressions and assigned a single syntactic category; for example the syntactic category for the words ”Andy Murray” is *Proper Noun*.

3.1.1 Framework

Fig. 3.1 shows the structure of the CGF framework, which consists of three phases: (1) grammar; (2) parsing and tagging; (3) learning and classification.

In Phase 1, a formal grammar (see Definition 1 in 3.2.1) is defined based on the analysis of the text in conjunction with the domain knowledge for a particular problem.

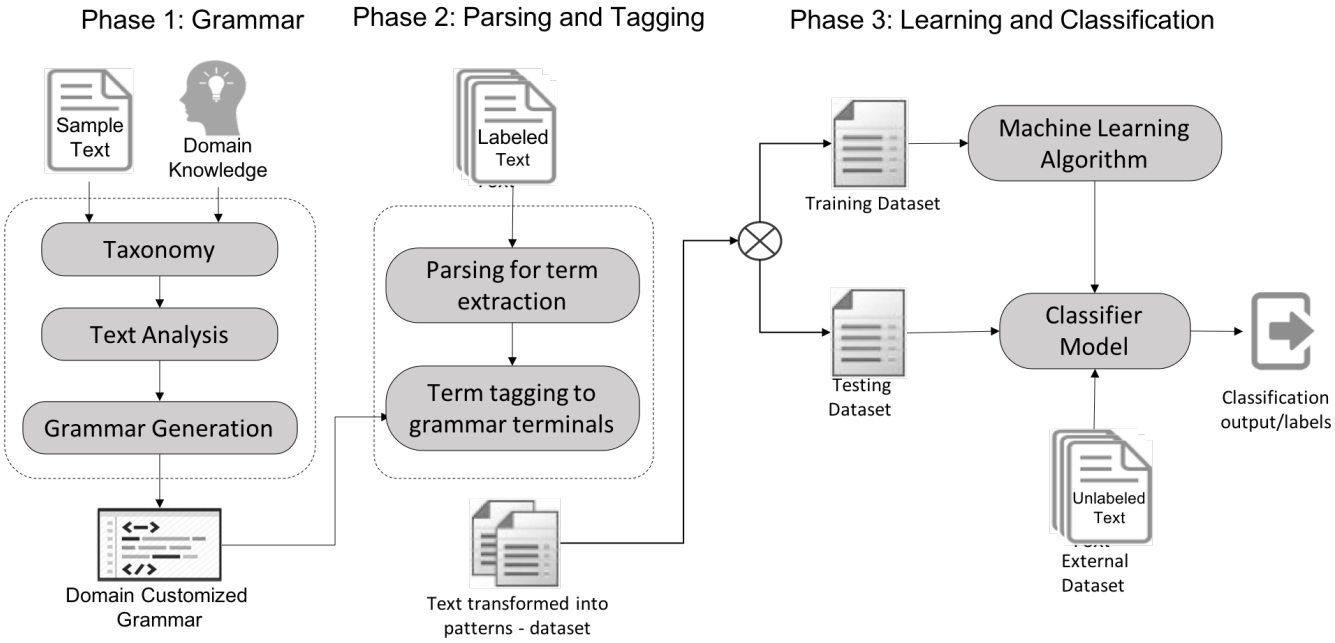

FIGURE 3.1 The figure shows the general CGF framework structure and the main three phases which are: (1) grammar; (2) parsing and tagging; (3) learning and classification

A taxonomy for a particular domain gives insight into the different characteristics of each category, by analysing examples of text from each taxonomy category, as well as using theoretical descriptions of these categories (from the documentation of the taxonomy), syntactic characteristics of each category can be identified. This, in turn, leads to the identification of particular characteristics that can be represented as domain-specific syntactic categories to be included in the terminals set of the grammar.

The grammar is used in Phase 2 to transform the text into syntactic patterns by first tokenizing the text into a series of non-terminal terms and then using the grammar production rules to parse the text and map the words to the grammar terminals. For example, the text *"Jane Austin books"* can be transformed into the pattern $[PN + CN]$, where *"Jane Austin"* has been mapped to PN (Proper Noun) and *"books"* has been mapped to CN (Common Noun).

After the labelled text has been transformed into syntactic patterns representation, Phase 3 takes place, in which a classification model is built by training a machine learning algorithm. The model can then be used for the classification of unlabelled text after transforming the unlabelled text into the syntactic patterns representation.

The use of the framework is illustrated in Chapter 4 and Chapter 5 for the problem of query and question classification.

3.2 The Customizable Grammar Framework

In this section, a detailed description of the CGF is presented. The three main phases of CGF are explained in detail in Sections 3.2.1, 3.2.2 and 3.2.3. Figure 3.2 shows more details of the three phases.

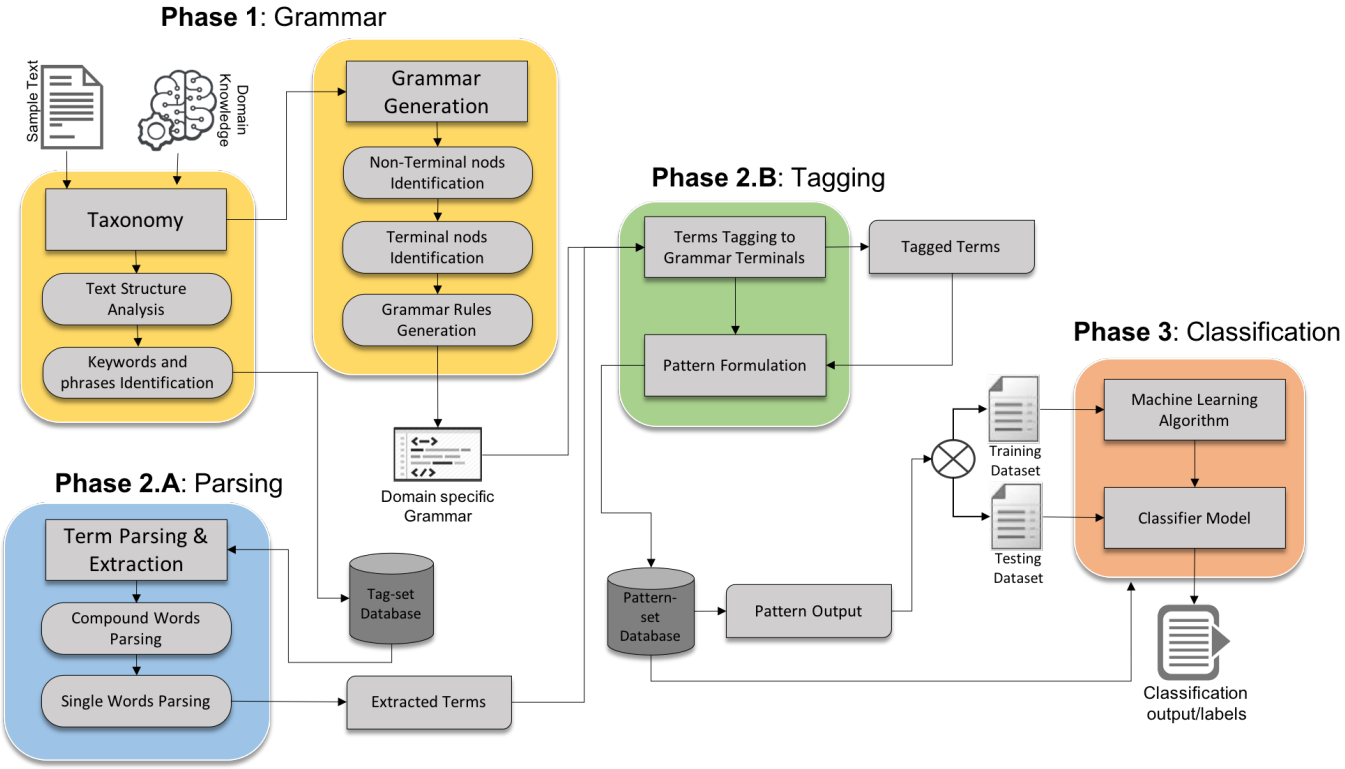

FIGURE 3.2 The figure shows in more detail the CGF framework structure and the main three phases which are: (1) grammar; (2) parsing and tagging; (3) learning and classification, in which phase (2) parsing and tagging is divided into two phases to show how these steps work.

3.2.1 CGF: Grammar

In this phase shown in Figure 3.2 Phase 1, input text is analysed using domain knowledge and a term taxonomy; this is done by identifying each keywords and phrases using the proposed tag-set (Section 3.2.2.1). Next, the grammar is generated by identifying terminal and non-terminals nods, the grammar in this phase is based on the Context-Free Grammar (CFG) which captures and combines two different components: the sentence structure and domain knowledge.

The context-free grammar is in the Backus Normal Form (BNF), even-though the BNF can not provide a full description of the English grammar [56], [103], the target is to use a

simple version of the English grammar combined with domain-specific syntactic categories since most domains do not perceive the formal English grammar and natural language.

DEFINITION 1 A grammar is a tuple (N, Σ, P, S), where:

1) N is a finite set of non-terminal symbols, which can be single words, such as *"Sport"*, or groups of words such as *"Paulo Coelho"* or *"Google Translate"*;

2) Σ is a finite set of terminal symbols that is disjoint from N (i.e Σ and N have no common elements); in our context the terminal symbols are syntactic categories (e.g. noun, verb, proper noun, action verb);

3) P is a finite set of production rules of the form $(\Sigma \cup N)^* N (\Sigma \cup N)^* \rightarrow (\Sigma \cup N)^*$, and

4) $S \in N$ is the starting symbol.

Creating grammatical rules helps in the identification of ambiguous terms since two different sentences may have similar terms but with different structures, each having a different meaning, which may lead to different intents.

For the examples *"Setup Instagram Application"* and *"Instagram Application Setup"* grammatical rule will be generated by identifying the structure of the sentences; (1) at phrase level, (2) at words level which includes word classes and sub-classes and (3) domain specific level.

A phrase, defined as a group of words that function as a single part of speech, can be a Verb phrase, Noun phrase, Determiner phrase, Adjective phrase, Adverb phrase or Prepositional phrase. Different classes of phrases contain different word classes. A word class or part of speech is a collection of words that can have sub-classes; the seven major word classes are Verb, Noun, Determiner, Adjective, Adverb, Preposition and Conjunction. Word order inside a phrase is one of the major structural ways in which the text can differ from each other. The position of a word depends on its word class, which means that each query could formulate a unique pattern.

At phrase level, *"Setup Instagram Application"* consists of *Verb Phrase* and *Noun Phrases*, while at word level, it consists of *Verb (Action Verb)* and *Nouns (Proper Noun and Common Noun)*. At the domain specific level it consists of *Action Verb - Interact* (AV_I), *Proper Noun - Software and Applications* (PN_{SA}) and *Common Noun - Other - Singular* (CN_{OS}).

On the contrary, at phrase level, *"Instagram Application Setup"* consists of *Noun Phrases*; at word level, *i*t consists of *Nouns (Proper Noun and Common Nouns)*. At the domain specific level it consists of *Proper Noun - Software and Applications (PN_{SA})* and *Common Noun - Other - Singular (CN_{OS})*.

The different syntactical structures of the two sentences leads to different syntactical patterns, which result in different meaning; intent and search results. Figure 3.3 illustrates in detail how a grammatical rule of a sentence is generated and how the domain specific grammar is created.

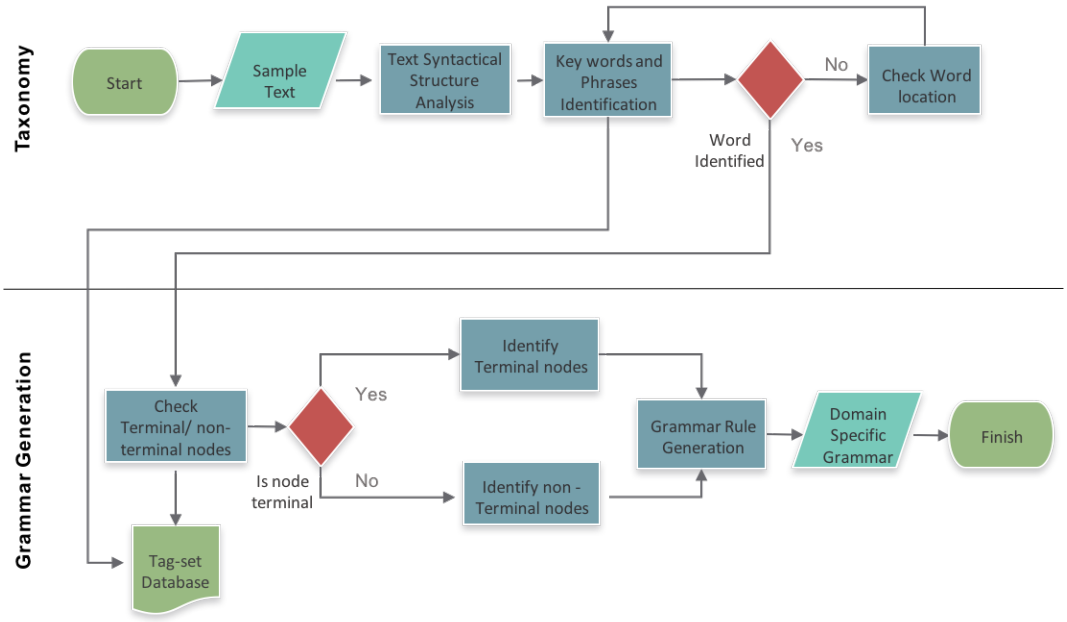

FIGURE 3.3 Phase 1: Grammar (Domain Specific Grammar Identification)

3.2.2 CGF: Parsing and Tagging

A syntax-based parsing and tagging [1] process is proposed using a grammar-based approach. This approach is a domain-specific approach, shown in Figure 3.2 Phase 2 and Phase 3, for the objective of assigning not just general PoS tags but also domain specific ones to help in the categorization and classification of text in different domains.

The aim of this approach is to create a simple parser and tagger that could easily be applied to different domains by creating domain specific grammatical rules, in which each text is transformed to general and domain specific PoS categories using these rules.

[1] subsequently, the term *Tagging* will also be referred to as *Mapping*

The grammatical rules contain in addition to typical categories of English grammar, domain-related grammatical categories. The domain-specific Syntax based parsing and tagging is described in the following sub-sections.

3.2.2.1 Tag-set

The tag-set was developed by [96]. It was mainly created for the purpose of identifying search queries by labelling each word in the query to its PoS and name entity to help in the classification of the users' intent. In this research, the tag-set was updated by adding more terms and categories.

The tag-set has been tested on different search engines' queries datasets, i.e. AOL 2006 data-set[2] [107] and the TREC 2009 Million Query Track data-set[3] [16].

Furthermore, it has been used in other domains such as question classification and also has been tested on different questions datasets, i.e. Yahoo Non-Factoid Question Dataset[4], TREC 2007 question answering data[5] and a Wikipedia dataset[6] that was generated by [125].

3.2.2.2 Tag-set categorization

The tag-set consists of 10,440 different words that have been labelled to PoS (Categories) which includes three levels of grammar taxonomy shown in Table 3.1; Level (1) which includes the seven major word classes in English, which are *Verb (V), Noun (N), Determiner (D), Adjective (Adj), Adverb (Adv), Preposition (P)* and *Conjunction (Conj)* in addition to *Question Words (QW)* ; level (2) consists of sub-categories of level (1) for example, *Common Nouns, Proper Nouns* and *Action Verbs*. In addition, the six main question words: *How, Who, When, Where, What* and *Which* have been added to this level. Level (3) which consists of all the domain specific categories for example, *Proper Noun Celebrity* and *Proper Noun Geographical Areas*. A list of all the syntactic categories and corresponding acronyms is displayed in Appendix A and Appendix B.

[2] http://www.researchpipeline.com/mediawiki/index.php?title=AOL_Search_Query_Logs
[3] http://trec.nist.gov/data/million.query09.html
[4] https://ciir.cs.umass.edu/downloads/nfL6/
[5] http://trec.nist.gov/data/qa/t2007_qadata.html
[6] https://www.cs.cmu.edu/~ark/QA-data

3.2.2.3 Constructing The Term Category Taxonomy

In order to construct term categories a random set of $100,000$ sample texts have been selected from the datasets that have been mentioned in section 3.2.2.1 .The following steps have been taken using a java program that has been developed by [96] for the mapping of each term to its word classes:

TABLE 3.1 The three levels taxonomy

Levels	Description
S	Consists of All Phrase classes
Level L1	Consists of the seven main word classes and question words
Level L2	Consists of the word classes sub-classes and the six main question words
Level L3	Consists of all domain specific classes

1) Parse the $100,000$ texts (queries/questions) and automatically extract terms.

2) Automatically map terms to their PoS tag, e.g. *"Capital of Spain"* is mapped as: (a) *"capital* $--> N$*"*, (b) *"of* $--> P$*"* and (c) *"Spain* $--> N$*"*.

   after tagging each term to one of the main word classes, a further tagging is done to assign each term to its sub-class if applicable. For example, the following terms will be tagged to (a) *"capital"* will be mapped to *"CN"*, (b) *"of* $--> P$*"* will not be mapped to any further categories and (c) *"Spain"* is mapped to *"PN"*.

3) Finally, after each term is mapped to one of the word classes or sub-classes, it will be mapped to the domain specific term category; the proposed categories were created after the analysis of the selected datasets. A detailed explanation of each category is provided in the appendix. For example, *"capital"* will be mapped to *"*CN_{OS}*"*, (b) *"of* $--> P$*"* will not be mapped to any further categories and (c) *"Spain"* is mapped to *"*PN_G*"*.

   The final step has resulted in the final refined taxonomy of term categories. The proposed tag-set contains all terms extracted from the dataset that have been used. In addition, all possible terms were added in all the seven main word classes except the Proper Noun Category, since Proper Nouns are infinite. Note that although the proposed so-

lution does not require knowing all Proper Nouns, it is still capable of classifying text that contain unrecognized Proper Nouns.

3.2.2.4 Parsing

This step is mainly responsible for extracting terms in the text. The system simply takes the text and parses it to help generate the grammar structure in the next phase to facilitate the tagging of each word to the right term category. Figure 3.4 illustrates in detail the parsing and terms extraction.

This phase is responsible for term parsing and extracting by using the keywords and phrases that have been identified from the previous phase; first compound words will be parsed and extracted then single words. Two examples are illustrated in Figure 3.5 for the sentences *University of Portsmouth Library'* and *Portsmouth Library'*. Figure 3.5 (1) illustrates the parsing of Compound word while Figure 3.5 (2) illustrates the parsing of single words.

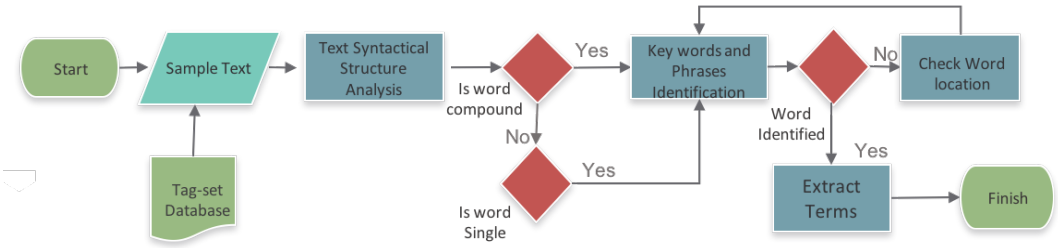

FIGURE 3.4 Phase 2A: Parsing (Terms Extraction)

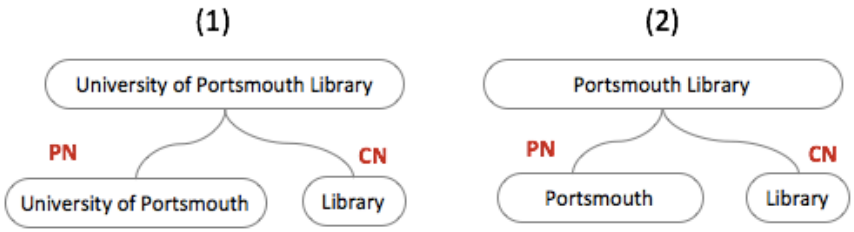

FIGURE 3.5 Example of parsing Compound and single words

3.2.2.5 Tagging

In this phase, the text is transformed into a pattern of grammatical terms by mapping each term to its grammar terminals; each term will be mapped to its highest level of abstraction

(word class, sub-class or domain specific) and after mapping each terms the syntactical pattern is formulated. Using domain specific grammar that has been generated from phase 1 (Grammar) terms will be tagged to its terminals in which tagged terms will be transferred to a pattern using the pattern-set.

An example is illustrated in Figure 3.6 for the given example, *"Capital of Canada"*. The figure illustrates the tagging of the terms to the grammar non-terminals. As a result of this process, the example is transformed into the following pattern: $[CN_{OS} + P + PN_G]$. In this given example the pattern is a representation of the most detailed syntactical pattern which is level 3.

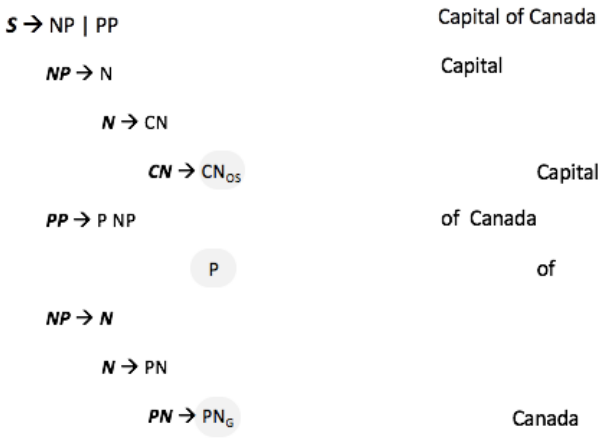

FIGURE 3.6 Example of how tagging is done

3.2.2.6 Features Representation

The proposed approach make use of two different features which are, grammatical features and domain specific features. Grammatical features have been used for the purpose of transforming the text (by using the grammar) into a new representation of grammatical terms, i.e. a syntactic pattern. In addition, it consists of other features such as singular and plural terms. Furthermore, domain-specific features (i.e. related to user-intent) were identified, which correspond to topics. Instead of further classifying the given text to fine grained or name entity, domain specific features were used to determine the type of the given text (user-intent). Hence the domain specific features contain less categories but still could identify the different user-intent.

Features are extracted and used for the classification and identification of the users' search intent. In Addition, as the length of the pattern varies depending on the structure of the given text, the number of the features varies. Hence, the number of attributes (features) in the dataset is equal to the size of the largest syntactic pattern as shown in Table 3.2.

TABLE 3.2 The table shows in detail the features representation of three different examples in which each user-intent consists of different feature representations (e.g. *"What is the smallest country in Africa?"* consists of seven features; Question word what QW_{What}, Linking verb LV, Determiner D, Adjective Adj, Common Noun Other Singular CN_{OS}, Preposition P, Proper Noun Geographical Areas PN_G.

Example (user-intent text)	Feature Representation						
	Feat. 1	Feat. 2	Feat. 3	Feat. 4	Feat. 5	Feat. 6	Feat. 7
What is the smallest country in Africa?	QW_{What}	LV	D	Adj	CN_{OS}	P	PN_G
Smallest country in Africa	Adj	CN_{OS}	P	PN_G	$Null$	$Null$	$Null$
Countries in Africa	CN_{OP}	P	PN_G	$Null$	$Null$	$Null$	$Null$

3.2.3 CGF: Learning and Classification

In this phase, the patterns that were generated in the tagging phase are used for machine learning, the aim of this phase is to build a model for automatic classification. The classification is done by following the standard process for machine learning, which involves the splitting of the dataset into a training dataset and a test dataset.

The training dataset is used for building the model, and the test dataset is used to evaluate the performance of the model. Once a model of satisfactory performance has been identified, it can be used for the classification of unlabelled text.

The machine learning algorithms that were used in this research are briefly described below. The *SVM* and *Naive Bayes* algorithms were used for the automatic classification due to the fact that they are among the most popular machine learning algorithms, and have also been popularly used in text classification tasks. Moreover, other classifiers such as *RandomForest*, *Decision tree (J48)* and *JRip* are now being used widely in text classification.

1) The Decision Tree (DT): is a method for approximating discrete-valued functions that is robust to noisy data and capable of learning disjunctive expressions. It classifies instances by sorting them down the tree from the root to some leaf node, which provides

the classification of the instance [89]. J48 and RandomForest are two of the widely used Decision Tree algorithms.

a) J48 Decision tree (J48): is an extension of the ID3 algorithm and is typically used in the machine learning and natural language processing domains [112]. The additional features of J48 are accounting for missing values, decision trees pruning, continuous attribute value ranges and derivation of rules. In the Weka data mining tool, J48 is an implementation of the C4.5 algorithm [113].

b) Random Forests (RF): are a combination of tree predictors in which each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest [11], [46].

2) Naive Bayes (NB): estimates the parameters of a multinomial generative model for instances, then finds the most probable class for a given instance using the Bayes' rule and the Naïve Bayes assumption that the features occur independently of each other inside a class [116]. In practice the Naïve Bayes learner performs remarkably well in many text classification problems [89] and is often used as a baseline in text classification because it is fast and easy to implement. Less erroneous algorithms tend to be slower and more complex [116].

3) In Support Vector Machine (SVM): input vectors are non-linearly mapped to a very high-dimension feature space. In this feature space a linear decision surface is constructed. Special properties of the decision surface ensures the high generalization ability of the learning machine [23]. SVMs are helpful in text categorization as their application can significantly reduce the need for labeled training instances in both the standard inductive and transductive settings. In addition, SVMs have the ability to generalize well in high-dimensional feature spaces. SVMs eliminate the need for feature selection making the application of text categorization considerably easier and do not require any parameter tuning since they can find good parameter settings automatically [51].

4) RIPPER rule learner (JRip): implements a propositional rule learner, Repeated Incremental Pruning to Produce Error Reduction (), as an optimized version of IREP. [21] [26]. In the Weka data mining tool, JRip is an implementation of the RIPPER algorithm.

3.3 Summary of Chapter

In this chapter, the Customizable Grammar Framework (CGF) was presented for the automatic classification of text through machine learning by taking advantage of domain-specific information and by preserving the structure of the text. For the later purpose, a new representation was proposed, in which text is represented as a syntactic pattern, i.e. a pattern formed of syntactic categories corresponding to the terms in the text. To transform the text into this representation a formal grammar-based approach was proposed. In addition, a syntax-based parsing and tagging process was proposed for the objective of assigning not just general PoS tags but also domain specific ones to help in the categorization and classification of text in different domains.

CHAPTER 4

Grammar-Based Framework for Query Classification

This chapter presents the Grammar Based Framework for Query Classification (GQC). An overview of the framework is presented in Section 4.1. Section 4.2 provides a detailed analysis of queries grammatical structure and full description of the different type of queries. Section 4.3 describes in detail the proposed query classification framework. The experiments setup and results are presented in Section 4.4, while the results are discussed in Section 4.5. Finally, Section 4.6 summarizes the chapter.

4.1 Overview

A Grammar Based Framework for Query Classification (GQC) is proposed, shown in Figure 4.1, GQC was adapted from CGF (chapter: 3). In order to make CGF compatible with the query classification problem, it was modified and adjusted, in which the tag-set, pattern-set and terms taxonomy were applied and used in a way that improved query identification; a further explanation will be provided in the following sections.

In addition, to identifying the relevant syntactic categories (both general and domain-specific), the different types of queries based on Broder's taxonomy [12] and Jansen's extended taxonomy [49] were analysed, as detailed in Sections 4.2.2 and 4.2.4, respectively. Based on the identified syntactic categories, the formal grammar is defined in Section 4.3.1.

The aim of GQC is to create a query identification and classification framework that could easily be applied by creating domain specific grammatical rules and patterns for each type of query. Query classification problem has been selected since search engines are the most

popular information retrieval application and query identification and classification play an important role in search engines and one of the major tasks in the enhancement of the classification process is the identification of query types.

The objectives of the research presented in this chapter are to:

1) Provide an analysis of web queries based on their syntactical structure.

2) Propose a framework that help in the identification of different query types.

3) Investigate the influence of the different levels of detail of domain-specific information (reflected in the domain-specific syntactic categories) on the classification performance;

4) Compare the performance of different machine learning algorithms for the classification of user intent;

5) Investigate the classification performance in comparison with state-of-the-art approaches.

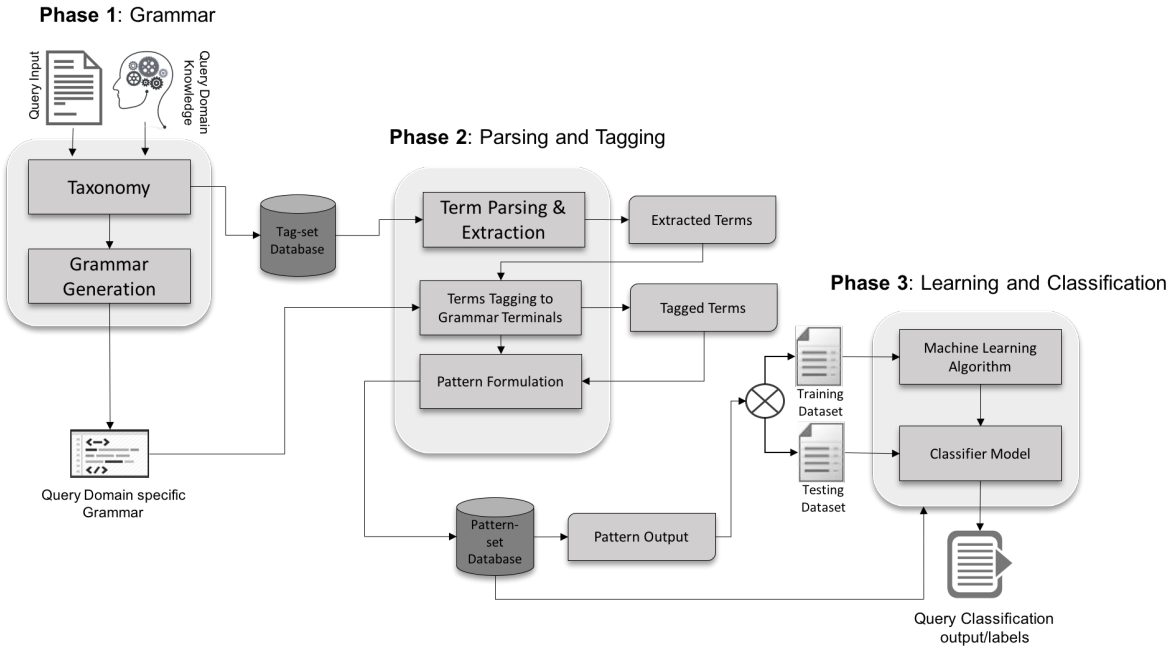

FIGURE 4.1 Query Classification Framework

4.2 Query Analysis

4.2.1 Queries Structure

Queries submitted to search engines are usually ambiguous and most of the queries might have more than one meaning, therefore using only the terms to identify search intents is not enough. To address this problem, the syntactic structure of the queries was explored.

Two different queries may have similar terms but with different structures, each having a different meaning, which may lead to different intents. For example, both queries *"George Orwell books order"* and *"order George Orwell books"* have similar terms and by just looking at them, one might assume that for both the intent is to buy books, i.e. transactional intent. According to the characteristics of the informational, navigational and transactional intents from [12], the first query is informational (i.e. find information on George Orwell books), while the second query is transactional (i.e. buy George Orwell books). Below is illustrated how the syntactical structure of the queries can reflect these different intents.

A phrase, defined as a group of words that function as a single part-of-speech, can be a Verb phrase, Noun phrase, Determiner phrase, Adjective phrase, Adverb phrase, Prepositional phrase or a combination of any of these phrases. Different classes of phrases contain different word classes. A word class or part-of-speech is a collection of words that can have sub-classes; the seven major word classes are Verb, Noun, Determiner, Adjective, Adverb, Preposition and Conjunction. Word order inside a phrase is one of the major structural ways in which the queries can differ from each other. The position of a word depends on its word class, which means that each query could formulate a unique pattern.

At phrase level, *"George Orwell books order"* consists of *Noun Phrases*, while *"order George Orwell books"* consists of a *Verb Phrase* and a *Noun Phrase*. At word level, *"George Orwell books order"* consists of *Nouns*, while *"order George Orwell books"* consists of a *Verb* and *Nouns*.

This different syntactical structure of the two queries leads to different syntactical patterns, which result in different meaning, intent and search results.

4.2.2 Analysis of Query Types (Broder's classification)

The characteristics of the three different types of queries were analysed, i.e. informational, navigational and transactional, from the point of view of the different word classes and types of phrases reflected in these queries. Details for each query type are given in the following sections.

4.2.2.1 Informational Query

One of the main feature that identifies the structure of informational queries is Phrases such as Noun phrase (NP), Verb phrase (VP), and Prepositional phrase (PP). For example *"location of apple stores in London"*.

The most used word class in this query type is Nouns, such as Common Nouns, e.g. *"county"*, *"company"* and *"place"*, and Proper Nouns, such as *"Spain"*, *"Eiffel Tower"* and *"The Beatles"*. Question words are also used; for example *"Why recycling is important?*; informational query is the only type of queries that contain Question words.

Moreover, queries in such search type could be short, medium or long in length, and they could contain one word or more than five words [49]. Furthermore, informational queries mostly formulate a complete sentence such as *"where can i buy vegan products in the UK?"*. However, in many cases, informational queries could be short in length [49], such as *Dinner ideas"*. Two examples of informational search syntactical structures are shown in Figure 4.2.

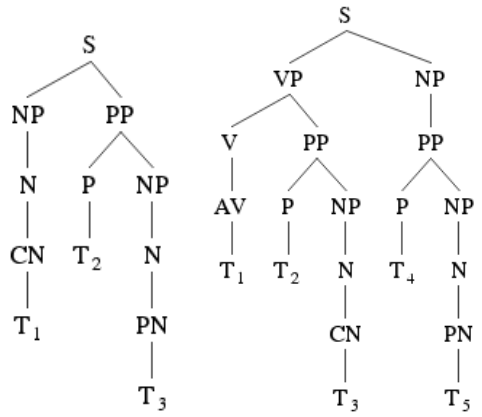

FIGURE 4.2 Examples of informational query structure using syntax tree representation, in which each sentence consists of a syntax structure of phrases (NP, PP, VP), word classes (N, V, P) and word sub-classes (PN, CN, AV); a sentence could have more than one of each.

4.2.2.2 Navigational Query

The structure of the query is the main feature that distinguishes navigational queries. This type of queries normally have a fixed syntactical structure which is the Noun Phrase (NP). Also, in some cases, the query contains a web-link or part of a web-link.

Furthermore, queries in this search type are mainly short, consisting of one or two words only [49]. Moreover, the only sub-class that could be found in this type of query is Proper Nouns since the query could contain just one word typically containing an organization, business, company or university name, such as *"Microsoft"*.

In addition, the structure of the query consists of domain suffixes and prefixes such as *"amazon.com"* and *"https://www.google.co.uk"*. Two examples of navigational search syntactical structures are shown in Figure 4.3.

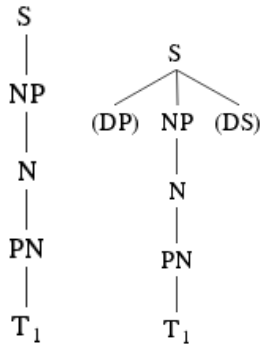

FIGURE 4.3 Examples of navigational query structure using syntax tree representation; the two patterns displayed cover the most common queries in the Navigational search. The sentences could consist of domain suffixes or prefixes (DS, DP), or have a syntactic structure of phrases (NP), word classes (N) and words sub-classes (PN).

4.2.2.3 Transactional Query

The syntactic structure of transactional queries consists mostly of Verb Phrases (VP) and Adjective Phrases (AP) for example *"buy cheap cars"*. Also, Noun Phrases (NP) could be in the structure of some queries – for example *"Celine Dion lyrics"*; however, some word classes are not used such as Question words, Pronouns, and Auxiliary verbs.

Moreover, most queries in transactional searching consist of Action Verbs (AV) such as *"order"*, *"buy"*, *"purchase"*, and *download"*. Furthermore, Adjectives are one of the word classes being used frequently in transactional queries, such as *"Free"* and *"online"*.

In addition, queries in this search type could be short or medium [49], they could contain one word or up to five words – for example *"cookie recipes"* and *"online pdf to word converter"*. Figure 4.4 shows two examples of transactional search syntactical structures.

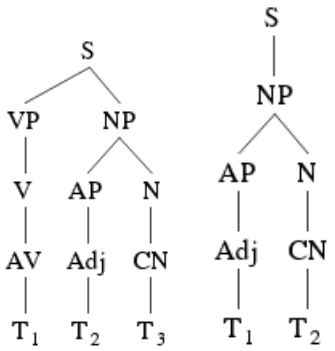

FIGURE 4.4 Example of a transactional query structure using syntax tree representation, in which each sentence consists of a syntactic structure of phrases (NP, AP, VP), word classes (N, V, Adj) and word sub-classes (CN, AV); a sentence could have more than one of each.

4.2.3 Analysis Overview for Broder's Classification

Based on the analysis above (section 4.2.2), an overview of the syntactical structure characteristics of the informational, navigational and transactional search type queries is presented in Tables 4.1, 4.2, 4.3 and 4.4.

Table 4.1 outlines the difference between the three types of queries from the point of view of word classes and Table 4.2 shows the types of phrases present in the three different query types. Both tables show that the navigational queries are clearly different from the other two, while the informational and transactional queries have a large similarity, indicating the difficulty in distinguishing them.

TABLE 4.1 Analysis of Word classes (Part-of-Speech) for Broder's Classification which include Word classes and the sentence length of Short (S), Medium (M) and Long (L)

Queries	Structure Length			Word classes							
	S	M	L	N	V	D	Adj	Adv	P	Conj	QW
Informational Query	√	√	√	√	√	√	√	√	√	√	√
Navigational Query	√	-	-	√	-	-	-	-	-	-	-
Transactional Query	√	√	-	√	√	√	√	√	√	√	-

TABLE 4.2 Analysis of Phrases for Broder's Classification

Queries	NP	VP	PP	AdvP	AdjP
Informational Query	√	√	√	√	√
Navigational Query	√	-	-	-	-
Transactional Query	√	√	√	√	√

Table 4.3 outlines the difference between the three types of queries based on different types of Verbs. Navigational queries do not typically contain Verbs, while the informational ones do. Moreover, the transactional queries tend to contain a particular type of Verb, i.e. Action Verb, but not the others, thus indicating that this particular Verb class plays an important role in the identification of transactional queries.

Table 4.4 outlines the different types of Nouns present in the three query types. Transactional queries tend not to include Pronouns, while the navigational queries typically do not include Common Nouns and Numeral Nouns.

TABLE 4.3 Breakdown Analysis of the Verb Class for Broder's Classification

Queries	AV	AuxV	LV
Informational Query	√	√	√
Navigational Query	-	-	-
Transactional Query	√	-	-

TABLE 4.4 Breakdown Analysis of the Noun Class for Broder's Classification

Queries	CN	PN	Pron	NN
Informational Query	√	√	√	√
Navigational Query	-	√	√	-
Transactional Query	√	√	-	√

4.2.4 Analysis of Query Extended Types (Jansen's Classification)

In this section, the analysis of the syntactic characteristics of the queries is described for Jansen's extended taxonomy [49].

4.2.4.1 Informational List:

Plural query terms (corresponding to the syntactic category Common Nouns Plural (CN_{OP})) are a highly reliable indicator of this type of query, since the goal is to find a list of suggested websites or candidates or a list of suggestions for further research, e.g. *"things to do in London"*. Word classes such as Common Nouns (CN) and Proper Nouns (PN) are mostly used, especially Common Nouns related to informational terms (CN_{Info}) such as list or play-list, and Entertainment terms (CN_{Ent}), such as Music, Movie, Sport, Picture, Game, e.g. *"list of Pixar movies"*. In addition, these queries include Proper Nouns terms related to products (PN_P), Geographical Areas (PN_G), Places and Buildings (PN_{PB}) and Institutions, Associations, Clubs, Parties, Foundations and Organizations (PN_{IOG}), e.g. *"London universities"*. In addition to the domain-specific syntactic categories mentioned, informational list queries also include general syntactic terms such as Action verbs (AV), Adjectives (Adj), Prepositions (P), Numeral Nouns (NN) and Determiners (D).

4.2.4.2 Informational Advice:

This type of queries consists mostly of: (a) Common Nouns terms related to ideas, suggestions, advice or instructions (CN_A), e.g. *"breakfast ideas"*; (b) question words such as how (QW_{how}) and what (QW_{what}), e.g. *"How to download iTunes"*; (c) Proper Nouns terms related to Software and Applications (PN_{SA}), such as *"itunes"*, *"Weka"* and *"Skype"*, Products (PN_P), such as *"iphone"* and *"Ben and Jerry's ice cream"*, Brand Names (PN_{BN}), such as *"Coach"*, *"Coca-Cola"* and *"Gucci"*. Furthermore, word classes such as Action Verbs (AV) and Numeral Nouns (NN) could be found in some queries.

4.2.4.3 Informational Find:

Since the goal of this category is to find or locate something in the real world like a product or service, the most used word sub-classes are Common Noun (*CN*) and Action Verb (*AV*), and especially terms related to find and locate (CN_L and AV_L). Moreover, Proper Noun terms like products (PN_P), Geographical Areas (PN_G), Places and Buildings (PN_{PB}) and Institutions, Associations, Clubs, Parties, Foundations and Organizations (PN_{IOG}) could be found in these queries since most product or shopping queries have the locate goal, e.g. *"Apple store location in New Jersey"* and *"cheap Apple MacBook pro"*. Furthermore, the only question word that is

used in this search type is *where* (WQ_{Where}) and is typically included in a complete sentence, e.g. *"Where is the location of Eiffel tower?"*.

4.2.4.4 Informational Undirected:

Most terms in this query are related to Proper Nouns such as terms related to science (PN_S), medicine (PN_{HLT}), history and news (PN_{HN}), and celebrities (PN_C), e.g. *"Michael Phelps"*, *"American Civil War"* and *"hypertension"*. Word sub-classes such as Common Noun (CN) and Numeral Noun (NN) are frequently used in this query type. Moreover, this is the only informational category that does not have some word classes such as Question words, Pronouns, Auxiliary verbs and linking verbs.

4.2.4.5 Informational Directed-Closed:

Queries in this category can be a question to find one specific or unambiguous answer, or to find information about one specific topic. Most queries in this type contain Common Noun terms related to Database and Servers (CN_{DBS}), such as Weather or Dictionary. In addition, they contain Proper Nouns terms related to Science (PN_S), Geographical Areas (PN_G), e.g. *"capital of Spain"*, Holidays, Days and Months (PN_{HMD}), such as *"Christmas"*, *"Monday"* and *"October"*. Furthermore, all question words such as when, how, where, what, who could be found in this search, e.g. *"What is a prime number?"*

4.2.4.6 Informational Directed-Open:

The structure of this category may take many forms; it might consist of either a question word such as How (QW_{How}), What (QW_{What}) and Why (QW_{Why}) to get an answer for an open-ended question, e.g. *"Why are gold valuable?"*, or it might consist of Common Nouns and Proper Nouns such as terms related to Science (PN_S) and Geographical Areas (PN_G) to find information about two or more topics, e.g. *"Insects communication"*.

4.2.4.7 Navigational Query:

These queries typically contain just Proper Nouns such as terms related to Company Names (PN_{CO}), Places and Buildings (PN_{BN}) and Institutions, Associations, Clubs, Parties,

Foundations and Organizations name (PN_{IOG}), such as *"IBM"*. In addition, the structure of the query consists of domain suffixes (DS) and prefixes (DP).

4.2.4.8 Transactional Interact:

These queries mainly consist of Action Verb and Common Noun terms related to interaction: (a) (AV_I), such as *Buy*, *Reserve* and *Order*, e.g. *"buy cell phones"*, and (b) (CN_I) such as *Translation* and *Reservation*. In addition, Common Nouns terms such as Database and Servers (CN_{DBS}), e.g. *"currency converter"*, *"stock quote" "weather"*, and File Type (CN_{file}), such as *MP3* and *PDF*, are highly used in this type of queries. Moreover, most transactional interact queries contain Proper Noun terms like Companies Name (PN_{CO}), Products (PN_P), Geographical Areas (PN_G), Places and Buildings (PN_{PB}), in addition to word class Adjective (Adj).

4.2.4.9 Transactional Download free:

Queries in this type of search mainly consist of Adjectives like *free* and *online* (Adj_F), (Adj_O), in addition to Action Verbs terms and Common Nouns terms related to download (AV_D), (CN_D), e.g. *"free online courses"* and *"free ebook downloads"*. They can also contain Common Noun terms, such as Entertainment (CN_{Ent}) and File Type (CN_{File}), as well as Proper Noun terms related to Software and Applications (PN_{SA}) and celebrity (PN_C).

4.2.4.10 Transactional Download not free:

These queries mainly consist of Adjectives (Adj), Action Verb terms and Common Nouns terms related to download (AV_D), (CN_D), e.g. *"lord of the flies book download"* and *"ABBA songs download"*. In addition, they contain Common Nouns terms such as Entertainment (CN_{Ent}) and File Type (CN_{File}), and Proper Noun terms related to Software and Applications (PN_{SA}) and products (PN_P).

4.2.4.11 Transactional obtain online:

This type of queries mainly consists of Common Noun terms related to obtained online (CN_{OO}), e.g. *"salmon recipes"*, Entertainment (CN_{Ent}), such as *"Sam Smith songs lyrics"*, in addition to Proper Nouns terms related to Celebrity (PN_C). Also, terms related to other

word classes and sub-classes such as Adjective (Adj) and Numeral Noun (NN) such as Ordinal Numbers (NN_O) and Cardinal Numbers (NN_C) could be in the structure of this type of query.

4.2.4.12 Transactional obtain offline:

This type of queries mainly consists of Common Noun terms related to obtain offline (CN_{OF}), e.g. *"Flowers wallpapers"* and *"Apple tv screensavers"*. In addition, it consists of adjective (Adj) terms, such as *free* (Adj_F), Proper Noun terms related to Software and Applications (PN_{SA}), Products (PN_P) and Celebrity (PN_C). Furthermore, word classes such as Linking Verbs (LV), Pronouns ($Pron$) and Auxiliary Verbs ($AuxV$) are not typically found in this query type.

4.2.5 Analysis Overview for Query Extended Classification

Based on the previous analysis (section 4.2.4), an overview of the syntactical structure characteristics of the extended classification of search type queries is presented in Tables 4.5, 4.6 and 4.7.

Table 4.5 outlines the difference between the twelve types of queries from the point of view of word classes, Table 4.6 outlines the difference between the twelve types of queries based on different types of Verbs and Table 4.7 outlines the different types of Nouns present in the twelve query types. Since navigational queries do not have an extended classification, the analysis of this type is similar to the one which was provided in Section 4.2.3

4.2.6 Query Terms Taxonomy

The following categories/word classes have been used, Verb (V), Noun (N), Determiner (D), Adjective (Adj), Adverb (Adv), Preposition (P) and Conjunction (Conj). In addition, question words (QW): how, who, when, where, what and which, were also used. Furthermore, two other classes were added: Domain Suffixes (DS) and Domain Prefixes (DP). Also, some word classes can have sub-classes. For example, Noun consists of sub-classes, such as Common Nouns (CN), Proper Nouns (PN), Pronouns (Pron) and Numeral Nouns (N); Verbs can be of several types, such as Action Verbs (AV), Linking Verbs (LV) and Auxiliary Verbs (AuxV).

TABLE 4.5 Analysis of Word classes (Part-of-Speech) for Query Extended Classification which includes Word classes and the sentence length of Short (S), Medium (M) and Long (L)

Queries	Structure Length			Word classes							
	S	M	L	N	V	D	Adj	Adv	P	Conj	QW
Info. List	√	√	√	√	√	√	√	-	√	-	-
Info. Advice	√	√	√	√	√	-	-	-	√	-	√
Info. Find	√	√	√	√	√	√	√	√	√	√	√
Info. Undirected	√	√	√	√	-	-	-	-	-	-	-
Info. Directed-Closed	√	√	√	√	√	√	√	√	√	√	√
Info. Directed-Open	√	√	√	√	√	√	√	√	√	√	√
Navi. Query	√	-	-	√	-	-	-	-	-	-	-
Tran.Interact	√	√	-	√	√	-	-	-	-	-	-
Tran.Download Free	√	√	-	√	√	-	√	-	-	-	-
Tran. Download not Free	√	√	-	√	√	-	√	-	-	-	-
Tran. Obtain online	√	√	-	√	-	-	√	-	-	-	-
Tran. Obtain offline	√	√	-	√	-	-	√	-	-	-	-

TABLE 4.6 Breakdown Analysis of the Verb Class for Query Extended Classification

Queries	AV	AuxV	LV
Info. List	√	-	-
Info. Advice	√	-	-
Info. Find	√	√	√
Info. Undirected	√	-	-
Info. Directed-Closed	√	√	√
Info. Directed-Open	√	√	√
Navi. Query	-	-	-
Tran.Interact	√	-	-
Tran.Download Free	√	-	-
Tran. Download not Free	√	-	-
Tran. Obtain online	-	-	-
Tran. Obtain offline	-	-	-

TABLE 4.7 Breakdown Analysis of the Noun Class for Query Extended Classification

Queries	CN	PN	Pron	NN
Info. List	√	√	-	√
Info. Advice	√	√	√	√
Info. Find	√	√	√	√
Info. Undirected	√	√	-	√
Info. Directed-Closed	√	√	-	√
Info. Directed-Open	√	√	-	√
Navi. Query	-	√	-	-
Tran.Interact	√	√	-	-
Tran.Download Free	√	√	-	-
Tran. Download not Free	√	√	-	-
Tran. Obtain online	√	√	√	√
Tran. Obtain offline	√	√	√	√

4.2.7 Constructing Query Term Taxonomy

The following steps have been taken for the analysis of the syntactic structure of the queries and the mapping of each term from a query to the word classes mentioned above, these steps are implemented using the same java program and procedures that have been mentioned in chapter 3, Section 3.2.2.3:

1) Parse and automatically extracting terms from each query.

2) Automatically map each term to its syntactic (PoS) word class; for example, in the query *"Who is Nikola Tesla"*, *"Who"* will be mapped to *"QW"*, *"is"* to *"LV"* and *"Nikola Tesla"* to *"PN"*.

3) Convert each query to its syntactical pattern. which is a representation of the original query where each term is replaced by a word class (PoS). For example, the query: "Free Wallpapers" is converted to the syntactical pattern: $[Adj + CN]$.

4) Categories each syntactical pattern into one of the search types (e.g Broder [12] or Jansen [49]).

4.3 Proposed Framework

The concept of the GQC is based on the use of grammar to capture and combine two different components: (a) sentence structure and (b) domain information. In order to achieve this, a customised grammar for the problem is developed. A context-free grammar in the Backus Normal Form (BNF) is used. As mentioned in chapter 3, it has been argued [56], [109], [103] that BNF can not provide a full description of the English grammar, however, the target is to use a simple version of the English grammar combined with domain-specific syntactic categories to guide the query classification stage.

4.3.1 Phase I: Grammar

In chapter 3, section 3.2.1 Definition 1 the formal grammar is defined as a tuple (N, Σ, P, S). In this section the details of the formal grammar are presented for the query classification domain.

The set N of non-terminals includes the terms in the queries, which can be single words, such as *"books"*, or groups of words such as *"Jane Austin"* or *"University of Portsmouth"*.

The set Σ of non-terminals consists of all the syntactic categories, both general and domain-specific. Table 4.8, reflecting five different levels of detail related to the syntactic categories; a list of all the syntactic categories and corresponding acronyms is displayed in Appendix A.

Below a number of rules are illustrated which show how the syntactic categories are derived, starting from the highest level (the starting symbol, i.e. the sentence/query) to the lowest level of detail (level 5).

$\langle S \rangle$ ::= $NP\langle S\rangle \mid VP\langle S\rangle \mid PP\langle S\rangle \mid AP\langle S\rangle \mid AdvP\langle S\rangle \mid NP \mid VP \mid PP \mid AP \mid AdvP$

$\langle NP \rangle$ ::= $N \mid D\ N \mid AP\ N \mid D\ AP\ N \mid P\ D\ N \mid A\ AP\ N \mid Adv\ P\ D\ N \mid Pron\ AP \mid Pron\ PP$

$\langle VP \rangle$ ::= $V \mid V\ PP \mid V\ NP \mid VP\ PP \mid AdvP\ VP \mid AuxV\ VP$

$\langle PP \rangle$ ::= $P \mid P\ NP \mid AdvP\ P\ NP \mid Adv\ P\ NP$

$\langle AP \rangle$ ::= $Adj \mid Adv\ Adj \mid Adj\ PP \mid Adj\ N$

$\langle AdvP \rangle$ ::= $Adv\ Adv$

$\langle NNP \rangle$ ::= $N\ PP \mid AP\ N \mid AP\ NN \mid NN\ PP \mid N\ PP$

$\langle V \rangle$ ::= $AV \mid LV \mid AuxV$

$\langle N \rangle$ ::= $PN \mid CN \mid NN \mid Pron$

$\langle QW \rangle$ ::= $Who \mid Where \mid What \mid When \mid Which \mid How$

$\langle AV \rangle$ ::= $AV_I \mid AV_L \mid AV_D$

$\langle CN \rangle$ ::= $CN_A \mid CN_{SWU} \mid CN_D \mid CN_{HN} \mid CN_{OS} \mid CN_{OP} \mid CN_I \mid CN_L \mid CN_{OB} \mid CN_{IFT}$

$\langle NN \rangle$ ::= $NN_C \mid NN_O$

$\langle PN \rangle$ ::= $PN_S \mid PN_{HLT} \mid PN_P \mid PN_{HMD} \mid PN_R \mid PN_{HN} \mid PN_{SA} \mid PN_{BN} \mid PN_E \mid PN_{Ent} \mid PN_{BDN} \mid PN_C \mid PN_G \mid PN_{IOG} \mid PN_{PB} \mid PN_{CO}$.

TABLE 4.8 Hierarchical structure of syntactic categories with different levels of details.

Levels	Description	Classes
S	Consists of All Phrase classes	$NP, VP, PP, AP, AdvP$.
Level 1	Consists of the seven main word classes and Question words	$N, V, Adj, Adv, Conj, D, P, QW$
Level 2	Consists of the word classes sub-classes	CN, PN, NN, $Pron$, AV, LV, $AuxV$, QW_{What}, QW_{Where}, QW_{When}, QW_{How}, QW_{Which}
Level 3	Consists of Level 2 specific sub-classes that were created for the query classification	Adj_{OF}, DS, DP, CN_O, CN_I, CN_L, CN_{OBEF}, CN_{EFI}, CN_D, CN_{HN}, CN_A, CN_{SWU}, CN_{DBS}, NN_C, NN_O, PN_{BBC}, PN_{HN}, PN_{HS}, PN_{HR}, AV_{IL}, AV_D
Level 4	Consists of Level 3 specific sub-classes that were created for the query classification	Adj_O, Adj_F, CN_{IFT}, CN_{Ent}, CN_{OB}, CN_{OO}, CN_{OS}, CN_{OP}, PN_{BSP}, PN_{CGIP}, PN_{BCEE}, PN_{HLT}, PN_S, PN_{HMD}, PN_R, AV_I, AV_L,
Level 5	Consists of Level 4 specific sub-classes that were created for the query classification	PN_{SA}, PN_{BN}, PN_E, PN_{Ent}, PN_{BDN}, PN_G, PN_{IOG}, PN_{PB}, PN_{CO}, PN_C, PN_P

4.3.2 Phase II: Parsing and Tagging

In Phase II, each query is parsed and mapped to the grammar terminals to transform it into a pattern of syntactic terms, as illustrated in Algorithm 1.

An example is illustrated in Fig. 4.5 for the query *'List of movies by Nicholas Sparks'*. The left-hand side of the figure illustrates the parsing of the query to extract the set of terms,

Algorithm 1 Parsing and Tagging Algorithm

Read query q from input file.
Read grammar rules and store it in G.
State Parse q and extract the set of terms T
for each t_i in T **do**
State c_i = Map(t_i, G) ▷ This maps term t_i based on G into category c_i
  if c_i is $null$ **then**
State $c_i = PN$ ▷ If no category found for term t_i, assume it is a Proper Noun.
    if c_{i-1} is PN **then**
State $combine(c_{i-1}, c_i)$ ▷ Replace any number of consecutive PN with a single PN
    end if
  end if
end for

while the right-hand side illustrates the mapping of the terms to the grammar non-terminals. As a result of this process, the example query is transformed into the following pattern: $[CN + P + CN + P + PN]$.

All queries are transformed into syntactic patterns through this process resulting into a dataset of labelled patterns. As the length of the pattern varies depending on the structure of the query, the number of attributes in the dataset is equal to the size of the largest syntactic pattern. In the datasets used for the experiments this maximum length was 13. For patterns of lower length, some attributes will have no values; for example, the pattern in the given example has a of length of 5, in which attributes 1 to 5 will have as values the syntactic categories from the pattern (i.e. CN, P, CN, P and PN) and the attributes from 6 to 13 will have no values.

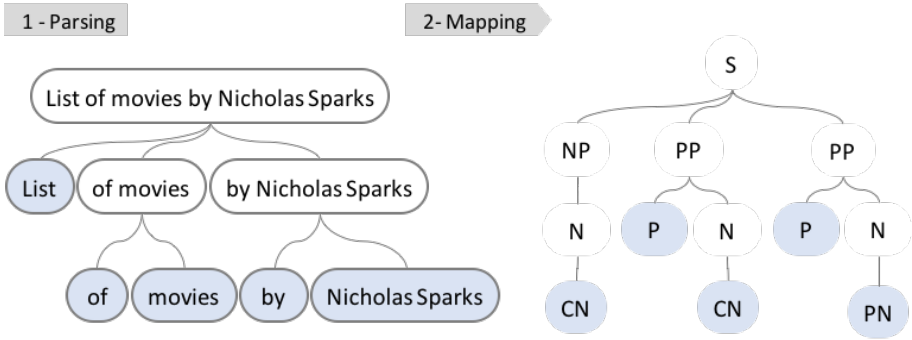

FIGURE 4.5 Phase II: Parsing and Tagging example

4.3.3 Phase III: Learning and Classification

In this phase, the patterns that were generated in Phase II are used for machine learning, with the purpose of building a model for automatic classification. The standard process for machine learning is followed, which involves the splitting of the dataset into a training dataset, which is used for building the model, and a testing dataset, which is used to evaluate the performance of the model. Once a model of satisfactory performance has been identified, it can be used for the classification of unlabelled queries.

Several learning algorithms were used and their performance was evaluated, as outlined in the Experiments section below.

4.4 Experiments

In this section, two sets of experiments were conducted to achieve the aims outlined in Section 4.1. For the first objective, i.e. investigate the influence of the different levels of detail of domain-specific information (reflected in the domain-specific syntactic categories) on the classification performance, experiments were conducted with different versions of the grammar, corresponding to the five levels for the terminals set; these experiments are described in sub-section 4.4.1. To validate the findings from the experiments related to the levels of detail for the grammar, another set of experiments were conducted, which are outlined in section 4.4.2.

For both sets of experiments, four machine learning algorithms were used: (1) decision trees, and in particular the J48 implementation in Weka; (2) RandomForest, (3) Repeated Incremental Pruning to Produce Error Reduction (RIPPER), and in particular the JRip implementation in Weka; (4) Naive Bayes.

The experiments were set up using the typical 10-fold cross validation and evaluation metrics, i.e. Accuracy, Precision, Recall and F-Measure. The classification of the queries were investigated according to Broder's categories (i.e. 3-class models), as well as Jansen's extended categories (i.e. 12-class models).

For the second objective, i.e. compare the performance of different machine learning algorithms for the classification of user intent, the experiment results will be analysed for both sets of experiments, as well as discussed overall. The third objective, i.e. investigate

the classification accuracy in comparison with state-of-the-art approaches, will be covered in Section 4.5, where the results are discussed of the proposed approach in comparison with previous ones.

TABLE 4.9 Data distribution

Query type		Frequency	Total
Informational			2980
	undirected	862	
	Advice	614	
	Directed - closed	642	
	Directed - open	127	
	Find	269	
	List	466	
Transactional			2220
	Download Free	42	
	Download not Free	49	
	Interact	420	
	Obtain Offline	383	
	Obtain Online	1326	
Navigational			684

4.4.1 Experiments on grammar levels

For this experiment, the 1953 labelled queries from [86] were used, and 4,047 queries were randomly selected from AOL 2006 dataset [107] and labelled according to the procedure described in [96]. From the 4,047 AOL queries, 116 were vague or contain mistakes and thus, were excluded, leading to 5,884 queries used in the experiments. Their distribution according to Broder's taxonomy and Jansen's extended taxonomy is given in Table 4.9.

The evaluation metrics for the 3-class models resulting from the four learning algorithms for each level of the grammar are displayed in Table 4.10. In addition to the overall performance, precision, recall and F-Measure are reported per class, to allow us to understand the effect of the additional syntactic categories per level on the identification of the three types of queries, i.e. informational, navigational and transactional.

The results show that with each level there is an improvement in the results, with signif-

icant improvements when moving from level 1 to level 2 and from level 2 to level 3. The improvement in performance from level 3 to level 4, and from level 4 to level 5, respectively, is marginal.

The results for the 12-class models are given in Table 4.11. These show similar results as for the 3-class models, with significant improvements form level 1 to level 2 and from level 2 to level 3. The improvement from level 2 to level 3 is higher than from the 3-class models, while the difference between level 4 and level 5 is marginal.

Level 1 and level 2 contain general syntactic categories of the English language. When only the higher level categories are used (i.e. level 1), while there are variations between the different learning algorithms, the overall picture is that the best performance occurs for informational queries, with the second best performance for transactional queries and the worst performance for navigational queries. In fact, three of the classifiers (GQC_{JRip}, GQC_{RF} and GQC_{J48}) are unable to identify navigational queries, and only the GQC_{NB} classifier is able to correctly identity some of navigational queries. These results show that based only on the syntactic categories at level 1, the machine learning algorithms are not able to distinguish well between the three types of queries, and are particularly unable to differentiate between the navigational queries and the other two types, i.e. informational and transactional.

When sub-categories of the English main syntactic categories are used, i.e. level 2, a dramatic improvement could be noticed in the performance of all classifiers in relation to navigational queries. In fact, all classifiers have a recall of 1 for this class, which indicates that there are no false positives, i.e. all instances identifies by the models as navigational are truly navigational. Also, the precision for all classifiers is above 0.9, indicating the presence of a small number of false positives, i.e. few informational or navigational queries are wrongly identified by the models as navigational. The sub-categories at level 2 have also marginally improved the performance for the informational and/or transactional queries for three classifiers (GQC_{RF}, GQC_{J48} and GQC_{NB}), while for GQC_{JRip} this improvement is more significant.

Level 3, which includes the first level of detail for the domain-specific syntactic categories, led to significant improvements of the performance of all classifiers for the informational and transactional queries; the performance for the navigational queries stayed the same as for level 2. These results indicate that the syntactic categories related to different domain-specific types of Common Nouns, Numeral Numbers, Proper Nouns, Adjectives and Action

Verbs, enable the machine learning algorithms to better differentiate between informational and transactional queries.

The performance of all classifiers for all classes improves further at level 4, which has more details related to the types of queries from Jansen's extended categories. There is an improvement even for the navigational queries, although there are no sub-types for the navigational queries in Jansen's extended categories, which indicates that some of the syntactic categories at level 4 enable the classifiers to better distinguish between the navigational queries on one hand, and the informational and transactional ones, on the other hand. In other words, the use of the level 4 syntactic categories lead to fewer false positives for the navigational class, i.e. fewer informational and transactional queries are mistaken for navigational ones. For the 12-class models (Table 4.11), the performance at level 4 shows a significant improvement compared with level 3, which is consistent with the fact that most of the syntactic categories from level 4 are derived from the analysis of Jansen's extended categories.

Finally, level 5 contains the most detailed level of domain-specific syntactic categories, related to aspects such as brand names, specific institutions and organisations, software, geographical areas, places and buildings, celebrity names and events. The use of these syntactic categories leads to further improvement for all classifiers and all classes, indicating that they enable the classifiers to better distinguish between the three types of queries.

In summary, the results show that using the domain-specific syntactic categories (levels 3, 4 and 5) leads to better classification performance compared with using standard English syntactic categories (level 1) and sub-categories (level 2). The results also indicate that the best performance is achieved when the most detailed domain-specific syntactic categories are used (level 5). This finding indicates that the grammar can be simplified by merging levels 3, 4 and 5 into one level, which would also simplify and speed-up the mapping in Phase II. To validate this new grammar structure, a new set of experiments were conducted, which is described in the next section.

4.4.2 Validation of the new grammar structure

The results from the previous experiments indicated that a simpler grammar structure with three levels would lead to a faster mapping process in Phase II. The new structure of the grammar with 3 levels is illustrated in Table 4.12. The new levels were denoted as L1, L2

TABLE 4.10 Performance of the classifiers Precision (P), Recall (R) and F-Measure (F) for Informational (Info.), Navigational (Nav.) and Transactional (Trans.) queries (3-class models)

		GQC_{JRip}			GQC_{RF}			GQC_{J48}			GQC_{NB}		
	Accuracy	55.11%			**66.26%**			66.02%			58.85%		
L1	Precision	0.53			0.85			0.84			0.87		
	Recall	0.94			0.69			0.69			0.53		
	F-Measure	0.68			0.76			0.76			0.65		
	Class	P	R	F	P	R	F	P	R	F	P	R	F
	Info.	0.53	0.94	0.68	0.84	0.69	0.76	0.84	0.69	0.76	0.87	0.53	0.66
	Nav.	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.27	0.10	0.15
	Trans.	0.71	0.20	0.31	0.53	0.83	0.65	0.53	0.83	0.65	0.48	0.83	0.61
L2	Accuracy	76.96%			**78.38%**			77.96%			71.59%		
	Precision	0.81			0.91			0.89			0.81		
	Recall	0.71			0.64			0.65			0.58		
	F-Measure	0.76			0.75			0.75			0.67		
	Class	P	R	F	P	R	F	P	R	F	P	R	F
	Info.	0.83	0.70	0.76	0.88	0.66	0.75	0.88	0.66	0.75	0.81	0.58	0.68
	Nav.	0.92	1.00	0.96	0.92	1.00	0.96	0.92	1.00	0.96	0.92	1.00	0.96
	Trans.	0.67	0.79	0.73	0.66	0.87	0.75	0.66	0.87	0.75	0.60	0.81	0.69
L3	Accuracy	98.47%			**98.67%**			98.47%			92.15%		
	Precision	1.00			1.00			0.99			0.93		
	Recall	0.98			0.98			0.98			0.92		
	F-Measure	0.99			0.99			0.99			0.92		
	Class	P	R	F	P	R	F	P	R	F	P	R	F
	Info.	1.00	0.98	0.99	1.00	0.98	0.99	0.99	0.98	0.99	0.93	0.91	0.92
	Nav.	0.92	1.00	0.96	0.92	1.00	0.96	0.92	1.00	0.96	0.92	1.00	0.96
	Trans.	0.99	0.99	0.99	1.00	0.99	0.99	1.00	0.99	1.00	0.91	0.90	0.90
L4	Accuracy	99.20%			**99.46%**			99.26%			88.64%		
	Precision	1.00			1.00			1.00			0.93		
	Recall	0.99			0.99			0.99			0.84		
	F-Measure	0.99			0.99			0.99			0.88		
	Class	P	R	F	P	R	F	P	R	F	P	R	F
	Info.	1.00	0.99	0.99	1.00	0.99	1.00	1.00	0.99	0.99	0.93	0.84	0.88
	Nav.	0.96	1.00	0.98	0.96	1.00	0.98	0.96	1.00	0.98	0.96	1.00	0.98
	Trans.	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.82	0.91	0.86
L5	Accuracy	99.62 %			**99.91%**			99.56%			89.21%		
	Precision:	1.00			1.00			1.00			0.93		
	Recall	1.00			1.00			1.00			0.85		
	F-Measure:	1.00			1.00			1.00			0.89		
	Class	P	R	F	P	R	F	P	R	F	P	R	F
	Info.	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.93	0.85	0.89
	Nav.	0.99	1.00	0.99	1.00	1.00	1.00	0.99	1.00	0.99	0.99	1.00	0.99
	Trans.	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.82	0.91	0.86

TABLE 4.11 Performance of the 12-class models. Precision (P), Recall (R), F-Measure (F).

	GQC_{JRip}				GQC_{RF}				GQC_{J48}				GQC_{NB}			
	Acc%	P	R	F	Acc%	P	R	F	Acc%	P	R	F	Acc%	P	R	F
L1	34.85	0.00	0.00	0.00	**48.66**	0.39	0.40	0.40	48.11	0.39	0.40	0.40	40.31	0.39	0.40	0.39
L2	52.88	0.84	0.02	0.04	**63.96**	0.51	0.24	0.32	63.15	0.51	0.23	0.32	52.75	0.47	0.25	0.33
L3	86.46	0.81	0.97	0.88	**90.16**	0.81	0.99	0.89	89.75	0.81	0.99	0.89	81.00	0.79	0.93	0.86
L4	96.50	0.99	1.00	0.99	**98.10**	1.00	1.00	1.00	97.38	0.99	0.99	0.99	91.41	0.95	0.94	0.95
L5	98.03	0.99	0.99	0.99	**99.16**	1.00	1.00	1.00	98.42	0.99	0.99	0.99	91.14	0.92	0.94	0.93

and L3 to distinguish them from the previous grammar structure denoted by levels 1 to 5.

TABLE 4.12 The three levels taxonomy

Levels	Description	Classes
S	Consists of All Phrase classes	$NP, VP, PP, AP, AdvP$.
Level L1	Consists of the seven main word classes and Question words	$N, V, Adj, Adv, Conj, D, P, QW$
Level L2	Consists of the word classes sub classes	$CN, PN, NN, Pron, AV, LV, AuxV$
Level L3	Consists of all the specific classes that were created for the query classification	AV_I, AV_L, AV_D, NN_C, NN_O, QW_Who, QW_{What}, QW_{Where}, QW_{When}, QW_{How}, QW_{Which}, DS, DP, PN_C, PN_S, PN_{HLT}, PN_{HMD}, PN_R, PN_{HN}, PN_{SA},PN_{BN}, PN_E, PN_{Ent}, PN_{BDN}, PN_G, PN_{IOG}, PN_{PB}, PN_{CO}, CN_A, CN_{SWU}, CN_D, CN_{HN}, CN_{OS}, CN_{OP} CN_I,CN_L, CN_{OB}, CN_{EFI}.

This modification results in the exclusion of 10 syntactic categories from levels 3 and 4 that contain sub-categories at levels 4 and 5, respectively. For example, the CN_{EFI} category at level 3 contains three sub-categories. In the merger, the CN_{EFI} category will be removed and its three sub-categories will become sub-categories of CN (from level 2). The same process is followed for all 10 syntactic categories that were removed. This results in a new level L3 that contains all the domain-specific syntactic categories as sub-categories of level 2 categories.

To validate this new grammar structure, experiments were conducted using the three levels and the same four machine learning algorithms. A new set of data of 8047 queries were randomly selected from the AOL 2006 dataset and labelled following the process used in [96]. These were used together with the 1953 labelled queries from [86] – thus, 10,000 queries were

used, which are distributed as outlined in Table 4.13.

TABLE 4.13 Data distribution

	Query type	Frequency	Total
Informational			5597
	undirected	1800	
	Advice	1018	
	Directed - closed	1042	
	Directed - open	259	
	Find	550	
	List	928	
Transactional			3012
	Download Free	48	
	Download not Free	65	
	Interact	696	
	Obtain Offline	502	
	Obtain Online	1701	
Navigational			1391

The results for the 3-class models are given in Table 4.14 and for the 12-class models in Table 4.15; the results per class using level L3 and RandomForest (GQC_{RF}) for the 12-class models are given in Table 4.16. As expected, the results for L1 and L2 are very similar to the results for levels 1 and 2 from the previous structure (displayed in Table 4.10), with slight variations which are likely due to the variation in the data used.

For level L3, the performance in similar to the results for level 5 in the previous structure (see Table 4.10), as both of these levels contain all the domain-specific syntactic categories. In the following, the results are discussed in relation to the objectives outlined in Section 3.1.

The Third objective was to investigate the optimal level of detail for the domain-related syntactic categories. The results from the experiments in Sections 4.4.1 and 4.4.2 indicate that the answer to this question is that the highest level of detail leads to the best classification performance. While the structure with 5 levels of details was very useful for understanding which syntactic categories influence the performance of the classifiers in relation to each type of query, the structure with the 3 levels is more useful for an automatic approach to query identification, facilitating a faster mapping process.

TABLE 4.14 Performance of the classifiers Precision (P), Recall (R) and F-Measure (F) for Informational, Navigational and Transactional queries (3-class models).

		GQC_{JRip}			GQC_{RF}			GQC_{J48}			GQC_{NB}		
L1	Accuracy	59.5%			**63.4%**			63.3%			53.71%		
	Precision	0.53			0.61			0.61			0.67		
	Recall	0.60			0.63			0.63			0.53		
	F-Measure	0.50			0.60			0.60			0.55		
	Class	P	R	F	P	R	F	P	R	F	P	R	F
	Info.	0.59	0.95	0.72	0.85	0.72	0.78	0.85	0.72	0.78	0.88	0.51	0.65
	Nav.	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.34	1.00	0.50
	Trans.	0.69	0.22	0.33	0.44	0.76	0.56	0.44	0.76	0.55	0.43	0.36	0.39
L2	Accuracy	76.3%			**77.8%**			77.6%			71%		
	Precision	0.77			0.81			0.80			0.76		
	Recall	0.76			0.78			0.78			0.71		
	F-Measure	0.76			0.78			0.78			0.71		
	Class	P	R	F	P	R	F	P	R	F	P	R	F
	Info.	0.82	0.75	0.78	0.89	0.70	0.78	0.88	0.70	0.78	0.85	0.59	0.69
	Nav.	0.91	1.00	0.95	0.91	1.00	0.95	0.91	1.00	0.95	0.91	1.00	0.95
	Trans.	0.61	0.68	0.64	0.61	0.83	0.70	0.61	0.82	0.70	0.52	0.80	0.63
L3	Accuracy	99.7%			**99.9%**			99.8%			95.5%		
	Precision	0.99			1.00			0.99			0.96		
	Recall	0.99			1.00			0.99			0.96		
	F-Measure	0.99			1.00			0.99			0.96		
	Class	P	R	F	P	R	F	P	R	F	P	R	F
	Info.	0.99	0.99	0.99	1.00	1.00	1.00	0.99	0.99	0.99	0.96	0.97	0.96
	Nav.	0.99	1.00	0.99	1.00	1.00	1.00	0.99	1.00	0.99	0.99	1.00	1.00
	Trans.	0.99	0.99	0.99	1.00	1.00	1.00	0.99	0.99	0.99	0.94	0.92	0.93

TABLE 4.15 Performance of the 12-class models.

	GQC_{JRip}				GQC_{RF}				GQC_{J48}				GQC_{NB}			
Levels	*Acc%*	*P*	*R*	*F*	*Acc%*	*P*	*R*	*F*	*Acc%*	*P*	*R*	*F*	*Acc%*	*P*	*R*	*F*
L1	30.5	0.21	1.00	0.35	**47.0**	0.44	0.41	0.42	46.7	0.44	0.41	0.42	38.6	0.44	0.41	0.42
L2	50.2	0.15	0.51	0.23	**63.7**	0.48	0.43	0.45	63.3	0.48	0.42	0.45	53.7	0.44	0.41	0.42
L3	99.2	0.99	1.00	0.99	**99.6**	1.00	1.00	1.00	99.3	1.00	1.00	1.00	92.0	0.91	0.94	0.93

TABLE 4.16 Performance of the 12-class RandomForest model by class for level L3.

Search Types	Precision	Recall	F-Measure
Informational undirected	1.00	1.00	1.00
Informational Advice	0.99	0.99	0.99
Informational List	0.99	1.00	0.99
Informational Directed Open	0.98	0.92	0.95
Informational Directed Closed	0.98	0.99	0.99
Informational Find	0.99	0.99	0.99
Navigational	0.99	1.00	1.00
Transactional Download Free	1.00	0.98	0.99
Transactional Download not Free	1.00	0.99	0.99
Transactional Interact	0.99	1.00	0.99
Transactional Obtain offline	0.99	1.00	0.99
Transactional Obtain Online	1.00	0.99	1.00

The Fourth objective was about which machine learning algorithms are best suited to classification of user intent, when using the data representation proposed in the GQC framework. Naive Bayes (GQC_{NB}), which is known to perform well on textual data, leads to the lowest performance models in the experiments (but not by much), while RandomForest (GQC_{RF}) leads to the best performing model. When using the domain-specific syntactic categories (levels 3, 4 and 5 in Tables 4.10 and 4.11, and level L3 in Tables 4.14 and 4.15) JRip (GQC_{JRip}) and J48 (GQC_{J48}) are very close in performance to RandomForest (GQC_{RF}), especially at level 5 in Table 4.10 and level L3 in Table 4.14. Consequently, the consistent performance of the classifiers validates the contribution of the new representation, with its domain-specific information and preservation of order, to the high classification performance.

The Fifth objective was about the classification performance of the proposed approach in comparison with state-of-the-art approaches. This is discussed in detail in the following section.

4.4.3 Performance comparison with other query classification approaches

For the objective of validating the proposed approach in improving the classification accuracy and the identification of different type of queries and to compare the classification performance of the proposed approach with the state-of-the-art approaches, experiments have been conducted using features classifier model based on the most used features in previous works such as n-gram in which $n = 2$, Bag-of-Words, Snowball Stemmer and stop words remover.

Similar to the previous experiments in section 4.4, to assess the performance of the machine learning classifier the experiments were set up using the typical 10-fold cross validation, i.e. the dataset is split into 10 folds, and each fold in used, in turn, for testing, while the other 9 are used for training. The output of the training process is a model, which is then used for classification in the test fold. The labels produced by the model are matched to the true labels and typical performance indicators, such as Accuracy, Precision, Recall, and F-Measure, are calculated. In addition, the following machine learning algorithms, were used for query classification. Which are; J48, RandomForests (RF) and Naive Bayes (NB).

4.4.3.1 Results

Table 4.17 presents classification performance details (Precision, Recall and F-Measure) of the n-gram$_{J48}$, n-gram$_{NB}$ and n-gram$_{RF}$ classifiers using broder's query categories. Results show that Decision Tree n-gram$_{J48}$ identified correctly (i.e. Recall) 90.9% of the queries, while n-gram$_{RF}$ identified correctly 95.2% of the queries and n-gram$_{NB}$, 95.2%.

TABLE 4.17 Performance of the classifiers using broder's categories and the features and n-gram framework - GQC_{RF} results are highlighted in bold. Precision (P), Recall (R), F-Measure (F).

	GQC_{RF}			n-gram$_{RF}$			n-gram$_{J48}$			n-gram$_{NB}$		
Accuracy:	**99.9%**			95.2 %			90.9%			95.2%		
Class:	P	R	F	P	R	F	P	R	F	P	R	F
Info.	**1.00**	**1.00**	**1.00**	0.94	0.93	0.93	0.82	0.95	0.88	0.94	0.93	0.93
Nav.	**1.00**	**1.00**	**1.00**	0.97	0.89	0.93	0.97	0.89	0.93	0.97	0.89	0.93
Tran.	**1.00**	**1.00**	**1.00**	0.96	0.99	0.97	0.96	0.89	0.93	0.96	0.99	0.97

In addition, Table 4.18 presents classification performance details (Precision, Recall and F-Measure) of the n-gram$_{J48}$, n-gram$_{NB}$ and n-gram$_{RF}$ classifiers using Jansen's extended query categories. Results show that Decision Tree n-gram$_{J48}$ identified correctly (i.e. Recall)

94.1% of the queries, while n-gram$_{RF}$ identified correctly 92.4% of the queries and n-gram$_{NB}$, 91.1%.

TABLE 4.18 Performance of the classifiers using Jansen's extended categories - GQC_{RF} results are highlighted in bold

	GQC_{RF}			n-gram$_{RF}$			n-gram$_{J48}$			n-gram$_{NB}$		
Accuracy:	**99.6%**			92.4%			94.1%			91.1%		
Class:	P	R	F	P	R	F	P	R	F	P	R	F
Info. undirected	**1.00**	**1.00**	**1.00**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Info. Advice	**0.99**	**0.99**	**0.99**	1.00	0.97	0.99	1.00	0.97	0.98	1.00	0.96	0.98
Info. List	**0.99**	**1.00**	**0.99**	1.00	0.95	0.97	1.00	0.95	0.97	0.98	0.95	0.96
Info. Directed Open	**0.98**	**0.92**	**0.95**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Info. Directed Closed	**0.98**	**0.99**	**0.99**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Info. Find	**0.99**	**0.99**	**0.99**	0.90	0.98	0.94	0.90	0.98	0.94	0.90	0.98	0.94
Nav.	**0.99**	**1.00**	**1.00**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Tran. Download Free	**1.00**	**0.98**	**0.99**	0.62	0.95	0.75	0.99	0.83	0.90	0.59	0.98	0.74
Tran. Download not Free	**1.00**	**0.99**	**0.99**	0.99	1.00	0.99	0.99	1.00	0.99	0.99	1.00	0.99
Tran. Interact	**0.99**	**1.00**	**0.99**	0.91	0.93	0.92	0.91	0.92	0.91	0.91	0.93	0.92
Tran. Obtain offline	**0.99**	**1.00**	**0.99**	0.97	0.47	0.63	0.64	1.00	0.78	0.99	0.47	0.64
Tran. Obtain Online	**1.00**	**0.99**	**1.00**	0.99	0.98	0.99	0.99	0.98	0.99	0.99	0.96	0.98

Even-though features such as n-gram, Bag-of Words, Snowball Stemmer and stop words remover could be used in the classification of informational, navigational and transactional queries, it could not be used in the classification of most extended categories. Informational queries extended categories such as undirected, directed- open and closed had 0 Precision, Recall and F-Measure for all the classifier. Similarly, navigational queries had 0 Precision, Recall and F-Measure for all the classifier. Furthermore, some transactional queries extended categories have low Precision and Recall such as transactional download free and transactional obtain-offline.

These results validate that using domain-specific information and preserving the structure of the query improve the classification accuracy and could be used in the identification of informational, navigational and transactional queries in addition to the extended categories of these queries.

4.5 Discussion

In this section, the performance of previous methods is discussed. The performance on previous automatic classification approaches is summarized in Table 4.19 (where several models are reported, e.g. with feature variations, the best performance was reported). With the exception of [49], which adopted a rule-based approach, all other approaches use machine learning. For [3] the values in the table are approximate numbers, as in the original paper they are displayed in a graph.

In terms of accuracy, the highest performance is obtained by [66], i.e. 90%, and [52]. A classification approach was used by [66] through linear regression, while [52] used a clustering approach through the k-means algorithm. Neither of these two works report performance by class. The proposed approach leads to over 99% accuracy overall, as well as very good performance by class, i.e. precision and recall values above 0.99.

Only two types of queries have been used by [66], i.e. informational and navigational; their argument for excluding the transactional category was the lack of agreement on this category, referred to as *resource* by [119] and as transactional by [12].

Based on their results [52], the authors argue that more refined categories such as the ones proposed by [119] and [49] may not be useful in practice because "they may not exhibit enough unique searching characteristics to permit this automatic classification" ([52], p.574). Their argument seemed to be supported by the low performance, i.e. 74% accuracy, of the rule-base approach in [49], in which characteristics of all refined categories were used for the identifications of informational, navigational and transactional queries. Most of the errors in the rule based approach by [49] were from the misclassification of navigational and transactional queries as informational. The results of the proposed approach (GQC) on the refined categories, i.e. the 12-class models (Table 4.16), however, indicate that it is possible to automatically classify them with very good levels of performance, i.e. precision and recall above 0.90.

Another approach that led to a relatively high performance is [86], which used three 2-class models, i.e. one for each type of query. They obtained overall F-values between 91% and 94%; they did not report results by class. The proposed approach (GQC) used one three-class model which outperforms each of the three 2-class models.

The majority of the previous approaches [3, 25, 34, 43, 44, 49, 77, 139] obtained better clas-

TABLE 4.19 Previous approaches performance [Algorithms (Alg), Accuracy (Acc), Precision (P), Recall (R)]

Reference	Alg	Acc	F-Measure				P	R	Notes
[66]	LR	90%							2 classes: informational and navigational
[77]	DR	80%	0.81				81.49	81.54	2 classes: C1=informational and transactional,
			C1	C2	C1		73.74	72.84	C2=navigational
			0.73	0.85	C2		85.62	86.18	
[3]	SVM				C1		0.7	0.9	3 classes: C1=informational,
					C2		0.55	0.4	C2=non-informational (navigational
					C3		0.35	0.2	and transactional), C3=ambiguous
[49]	rules	74%							most errors are from misclassifying navigational and transactional queries as informational
[2]	SVM	84.5%			C1		0.86	0.87	2 classes: C1=navigational and
					C2		0.81	0.80	C2=informational
[86]	SVM		91–94%						three 2-class models: informational/other; navigational/other; transactional/other;
[52]	k-means	94%							8 clusters: 6 navigational; 1 transactional and 1 navigational
[44]	SVM		94.87				94.87	94.87	2 classes: navigational, informational
	SVM		79.18				79.18	79.18	3 classes: navigational, informational, transactional
[34]	SVM		0.4594				0.8238	0.4463	2 classes: C1=informational and C2=non-
			C1	C2	C1		0.7227	0.9915	informational (transactional and navigational)
			0.82	0.68	C2		0.8917	0.2948	
[43]	NB		C1	C2	C3	C1	0.929	0.886	3 classes: C1=informational, C2=transactional,
			0.86	0.82	0.39	C2	0.84	0.810	C3=navigational
						C3	0.275	0.698	
	SVM		C1	C2	C3	C1	0.867	0.983	
			0.92	0.80	0.00	C2	0.795	0.810	
						C3	0.00	0.00	
[139]	SVM	64.4%							2 classes: navigational and informational
[25]	MaxEnt	82.22%				C1		88.23	3 classes: C1= informational, C2=navigational,
						C2		79.42	C3=resource/transactional
						C2		66.56	
	SVM	78.68%				C1		89.16	
						C2		70.96	
						C3		65.83	
	NB	81.41%				C1		86.38	
						C2		77.59	
						C3		76.21	

sification results for the informational queries compared with navigational and transactional ones, leading to two different approaches to this problem: (a) eliminating the transactional category [2, 66, 139]; (b) merging some categories, e.g. informational with transactional [77], navigational with transactional [3, 34]. Some found the transactional ones more difficult to identify than the navigational ones [25], while others found the opposite [43].

Without the domain-specific syntactic categories (i.e. levels 3, 4, 5 and L3), the results of (GQC) had the same tendency as the ones in [25], i.e. navigational queries were more easily identified than transactional ones. This may be due to the use of similar features which focus on detailed linguistic information, unlike [43], who used some linguistic information such as specific transactional and interrogative terms (corresponding to transactional and informational queries), but little specific information about navigational queries.

In conclusion, The proposed approach (GQC) outperforms the previous ones due to the use of domain-specific information and the preservation of structure in query representation, while also having practical advantages related to the reduced number of features, and an automatic grammar-based approach for transforming queries into the syntactic patterns representation.

4.6 Summary of Chapter

In this chapter, the Customizable Grammar Framework (CGF) for user intent text classification was applied to query classification problem in which the Grammar Based Framework for Query Classification (GQC) was introduced with the objective of creating a query identification and classification framework that could easily be applied by creating domain specific grammatical rules and patterns for each type of query. In addition, general and domain-specific syntactic categories were identified and different types of queries were analysed. Moreover, experimental results showed that the proposed approach outperforms previous ones, both overall, as well as for each type of query. In addition, the proposed approach addressed one of the major issues in text representation, i.e. large sparse datasets, by requiring a significantly smaller number of features.

CHAPTER 5

Grammar-Based Framework for Question Categorization and Classification

This chapter presents the Grammar-Based Framework for Question Categorization and Classification (GQCC). First, an overview of the framework is presented in Section 5.1. Section 5.2 provides a detailed analysis of the questions grammatical structure and full description of the different types of questions. Section 5.3 describes in detail the proposed question classification framework. The experiments setup and results are presented in Section 5.4, while the results are discussed in Section 5.5. Finally, Section 5.6 summarizes the chapter.

5.1 Overview

A Grammar Based Framework for Question Categorization and Classification (GQCC) was proposed, shown in Figure 5.1, GQCC was adapted from CGF (chapter: 3). In order to make CGF compatible with the question categorization and classification problem, it was modified and adjusted, in which the tag-set, pattern-set and terms taxonomy were applied and used in away that enhanced question identification. A further explanation will be provided in the following sections.

The aim of GQCC is to create a question categorization and classification framework that could easily be applied to different question-answering systems by creating domain specific grammatical rules and patterns for each type of question. Questions Classification (QC) problem has been selected since question-answering has become one of the most popular informa-

tion retrieval applications and QC plays an important role in question-answering systems and one of the major tasks in the enhancement of the classification process is the identification of questions types.

GQCC transforms the question using grammatical rules into a new form of representation in which each term in the question is represented as its grammatical category, which is called syntactical pattern, which has the advantage of preserving the grammatical structure of the question. The grammatical rules contain in addition to typical categories of English grammar, domain-related grammatical categories. Furthermore, in order to transform the question into a syntactical patterns a formal grammar approach is used and a machine learning is applied on this transformed data to obtain models for automatic classification.

The objectives of the research presented in this chapter are to:

1) Provide an analysis of question types based on their syntactical structure.

2) Propose a framework that help in the identification of different question types.

3) Investigate the impact of using different levels of detail of grammatical categories and domain-specific information on the classification performance.

4) Compare the performance of different machine learning algorithms for the classification of question intent;

5) Investigate the classification performance in comparison with state-of-the-art approaches.

5.2 Question Analysis

5.2.1 Analysis of Questions Structure and Characteristics

A new question categorization is proposed which is based on the general question types. The objective of this classification is to focus on the general and simple type of questions that are asked by most people. This classification is motivated by the basic English grammar [68], [35] and by the categorization of questions in [13, 14, 58]. After the analysis of different datasets, i.e. Yahoo Non-Factoid Question Dataset[1], TREC 2007 Question Answer-

[1] https://ciir.cs.umass.edu/downloads/nfL6/

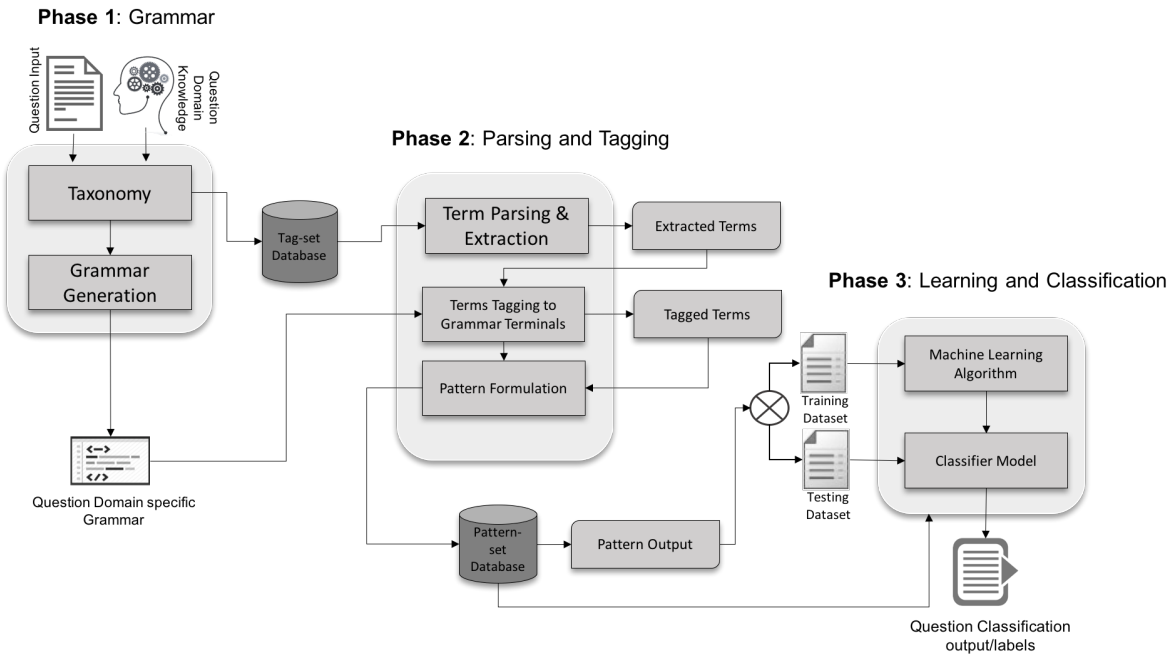

FIGURE 5.1 Question Classification Framework

ing Data[2] and a Wikipedia dataset[3] that was generated by [125], questions were classified into six different categories, which are: causal, choice, confirmation (Yes-No Questions), factoid (Wh-Questions), Hypothetical and list. These categories are based on the question types in English and the classification is based on types of questions asked by users and the answers given. Each of these questions has its own characteristics, features, and structure that help in the identification of each type. The choice, confirmation (Yes-No Questions), factoid (Wh-Questions) and Hypothetical questions were adapted from the English grammar, while list and causal were adapted from previous works. Table 5.1 shows a summary of the question types structure and characteristics which are detailed below.

1) *Yes-No Questions (Confirmation Questions):* This type of question begins with an auxiliary verb or linking verb, and the expected answer is either 'Yes' or 'No', for example *"Is Detroit a city in Michigan?"*. In addition, the question could start with negative auxiliary verbs or linking verbs, such as *"Wasn't Leonardo da Vinci born on April 15?"*. Moreover, this type of question usually does not contain a question word.

2) *Wh-Questions (Factoid Questions):* The main feature of this type of question is the presence of question words, e.g. *"What did Alessandro Volta invent in 1800?"*; any

[2] http://trec.nist.gov/data/qa/t2007_qadata.html
[3] https://www.cs.cmu.edu/~ark/QA-data

TABLE 5.1 Question Types Structure and Characteristics

Questions	Answer	PoS that identify the Question
Confirmation	Yes or No	AuxV
Factoid	Any kind of information could be given as an answer	QW
Choice	A selection between two or more options	Conj (OR) , LV, AuxV
Hypothetical	Any kind of information could be given and could have more than one accurate answer	QW (What)
Causal	Deep explanations and elaborations related to the topic in the question	QW (Why, How)
List	A list of different Facts, Entities, Events and Names, depend on the topic.	Plural (CN), QW (What, Which, Who)

kind of information can be given as a response. Furthermore, most of them start with a question word, such as *What / Where / Why / Who / Whose / When / Which*. However, there are other question words that do not start with "wh" as well, e.g. *how / how many / how often / how far / how much / how long / how old*. In addition, the structure of the question could begin with a Preposition followed by a question word, *"P + QW"*, rather than a question words, such as *"In what year was Nairobi founded?" / "At what distance does the earth curve?"*. Also in many cases the question word could be found in the middle of the question, for example *"Water boils at what temperature?"*. Most factoid questions are formulated as an advice question, e.g. *"How do you quit smoking?"*, and are related to facts, current events ideas and suggestions. In addition, some factoid questions could contain two types of questions, factoid and causal, for example *"What is a good phone service and why?"*.

3) *Choice Questions:* The structure of this type of question mainly offers choices in the question; usually the question contains two (or more) presented options. These options are connected using the conjunction "OR". Questions in this type could begin with a: (a) linking verb, e.g. *"Was ancient Egypt before or after ancient Greece?"*; (b) Auxiliary Verb, e.g. *Did Einstein die in the 50s or 60s?*; (c) Question word, e.g. *"What is better Samsung or iPhone?"* or (d) Determiner, e.g. *"Which is better Netflix or*

Amazon?". Furthermore, some choice questions could contain causal questions, For example, *"Which is better Playstation or Xbox 360 and why?"*

4) *Hypothetical Questions:* A hypothetical question is asked to have a general idea of a certain situation. The question typically begins with the question word *"What"*, e.g. *"What would you do if someone had a heart attack?"*; *"What would happen if the nervous system stopped working"*. It is mainly a what/if question.

5) *Causal Questions:* The structure of this question begins with the question words *"How"* or *"Why"* and the answer requires further explanation; for example, *"Why do clouds turn dark when it's about to rain?"*. However, the question could begin with *"if"*, and takes the following format *"if...then...why"* or *"if...then...how"*. In addition, causal questions could in many cases begin with a question word followed by a negative linking verb or a negative auxiliary verb, for example *"Why isn't my phone connecting to wifi?"*.

6) *List Questions:* The answer of this type of question takes the form of a list of entities or facts. Plural terms are a highly reliable indicator of this question. In addition, this question often begins with the words *"List"* or *"Name"* (e.g. *"List of Disney movies" / "Name of dinosaurs"*) or a question word followed by a plural term, such as *"What countries are in Europe?"*, *"Which products contain gluten?"*. However, in some cases list questions could begin with a preposition followed by a question word, for example *"In what countries does Uber operate?"/ "In which African countries is French spoken?"*.

5.2.2 Validation of Questions Types Categories

A validation set was created by having three annotators independently judge 200 questions that were randomly selected from the sample of 5,000 that was obtained from the three data-sets mentioned previously: Yahoo Non-Factoid Question Dataset , TREC 2007 Question Answering Data and a Wikipedia dataset that was generated by [125].

Questions were labelled by assessors according to the categorization of questions that was discussed in Section 5.2.1. In the first stage, two annotators labelled the questions, and then the classification results were reviewed . If a question was labelled differently by the two

annotators, a third annotator assigned a label to the question. The two annotators disagreed on 5.5% of the questions.

5.2.3 Question Terms Taxonomy

The terms taxonomy has been used for the purpose of transforming the questions (by using the grammar) into a new representation as a series of grammatical terms, i.e. a syntactical pattern.

The terms taxonomy is mainly based on the seven major word classes in English, which are Verb (V), Noun (N), Determiner (D), Adjective (Adj), Adverb (Adv), Preposition (P) and Conjunction (Conj). In addition, a category for question words (QW) were added that contains the six main question words: ”how”, ”who”, ”when”, ”where”, ”what” and ”which”. Some word classes like Noun can have sub-classes, such as Common Nouns (CN), Proper Nouns (PN), Pronouns (Pron) and Numeral Nouns (NN), as well as Verbs, such as Action Verbs (AV), Linking Verbs (LV) and Auxiliary Verbs (AuxV). In addition to the English grammar terms, domain-specific terms (i.e. related to question-answering) were identified, which correspond to topics – these are listed in Table 5.2.

In Table 5.3 the three different levels of detail related to the grammatical categories are presented to enable us to establish the influence of the different levels of detail on the classification performance; a list of all the grammatical categories and corresponding acronyms is displayed in Appendix B.

5.2.3.1 Constructing Question Term Taxonomy

In order to construct term categories the following steps have been taken, these steps are implemented using the same java program and procedures that have been mentioned in chapter 3, Section 3.2.2.3:

1) Parse and automatically extract terms from each question.

2) Automatically map terms to their PoS tag, e.g. *”Where is the city of Bath”* is mapped as: *”Where* $->$ *QW”*, *”is* $->$ *LV”*, *”the* $->$ *D”*, *”city* $->$ *N”*, *”of* $->$ *P”* and *”Bath* $-->$ *N”*.

after tagging each term to one of the main word classes mentioned above, a further tagging is done to assign each term to its sub-class if applicable. For example, *"Where"*, *"is"* and *"the"* will not be mapped to any further categories, *"city"* will be mapped to *"CN"*, *"of"* will not be mapped to any further categories and *"Bath"* will be mapped to *"PN"*.

3) Finally, after each term is mapped to one of the word classes or sub-classes, it will be mapped to the domain specific term category; the proposed categories were created after the analysis of the selected datasets. A detailed explanation of each category is provided in the appendix. For example, *"What"* will be mapped to $-> QW_{where}$", *"is"* and *"the"* will not be mapped to any further categories, *"city"* will be mapped to *"*CN_{OS}*"*, *"of"* will not be mapped to any further categories and *"Bath"* will be mapped to *"*PN_G*"*.

TABLE 5.2 Domain Specific Terms Categories

Category Name	Terms Example
Health	Specific Terms related to health, medicine, beauty.
Sports	Game and recreation, sports events, sports.
Arts and entertainment	Entertainment, Celebrities Name, lyrics, Movies, Books, Authors
Food and drinks	Foods, Drinks, Recipes
Animals	Pets, wild animals.
Science and math	Specific Terms related to Science and math.
Technology and internet	Software and Applications, Site, Website, URL, Database and Servers.
Society and culture	Environment, Holidays, Months, history, political, Relationships, Family.
News and events	Newspapers, Magazines, Documents, Events.
Job, Education and Reference	Careers, Institutions, Associations, Clubs, Parties, Foundations and Organizations.
Business and Finance	Money, company, products, Economy.
Travel and places	Geographical Areas, Transportation, Places and Buildings, Countries.

TABLE 5.3 The three levels taxonomy

Levels	Description	Classes
S	Consists of All Phrase classes	$NP, VP, PP, AP, AdvP$.
Level L1	Consists of the seven main word classes and Question words	$N, V, Adj, Adv, Conj, D, P, QW$
Level L2	Consists of the word classes sub-classes	$CN, PN, NN, Pron, AV, LV, AuxV$
Level L3	Consists of all the specific classes that were created for the question classification	NN_C, NN_O, QW_{Who}, QW_{What}, QW_{Where}, QW_{When}, QW_{How}, QW_{Which}, PN_C, PN_S, PN_{HLT}, PN_{HMD}, PN_R, PN_{HN}, PN_{SA}, PN_{BN}, PN_E, PN_{Ent}, PN_{BDN}, PN_G, PN_{IOG}, PN_{PB}, PN_{CO}, CN_A, CN_{SWU}, $CN_{HN}, CN_{OS}, CN_{OP}, CN_{HLT}$.

5.3 Proposed Framework

The proposed GQCC framework makes use of two related sources of information about the questions, i.e. the structure of different questions and question domain-specific information available about each category of questions. In order to capture the relation between these two sources and combine them in a unified structure, a formal grammar is designed for the question classification problem. As mentioned in the previous chapters (3, 4) the context-free grammar in the Backus Normal Form (BNF) is adopted in this study as it is the most widely used grammar in computing and the target in this research is to use a simple version of the English grammar combined with domain-specific grammatical categories to guide the question classification and categorization stage.

5.3.1 Phase I: Grammar

In chapter 3, section 3.2.1 Definition 1 the formal grammar is defined as a tuple (N, Σ, P, S). In this section, the details of the formal grammar are presented for the question classification domain. Below a number of rules are illustrated which show how the grammatical categories are derived, starting from the highest level (the starting symbol, i.e. the question) to the lowest level of detail (level 3).

$\langle S \rangle$ ::= $NP\langle S \rangle \mid VP\langle S \rangle \mid PP\langle S \rangle \mid AP\langle S \rangle \mid AdvP\langle S \rangle \mid NP \mid VP \mid PP \mid AP \mid AdvP$

$\langle NP \rangle$ ::= $N \mid D\ N \mid AP\ N \mid D\ AP\ N \mid P\ D\ N \mid A\ AP\ N \mid Adv\ P\ D\ N \mid Pron\ AP \mid Pron\ PP$

$\langle VP \rangle$ ::= $V \mid V\ PP \mid V\ NP \mid VP\ PP \mid AdvP\ VP \mid AuxV\ VP$

$\langle PP \rangle$ ::= $P \mid P\ NP \mid AdvP\ P\ NP \mid Adv\ P\ NP$

$\langle AP \rangle$ ::= $Adj \mid Adv\ Adj \mid Adj\ PP \mid Adj\ N$

$\langle AdvP \rangle$ ::= $Adv\ Adv$

$\langle NNP \rangle$ ::= $N\ PP \mid AP\ N \mid AP\ NN \mid NN\ PP \mid N\ PP$

$\langle V \rangle$ ::= $AV \mid LV \mid AuxV$

$\langle N \rangle$ ::= $PN \mid CN \mid NN \mid Pron$

$\langle QW \rangle$ ::= $Who \mid Where \mid What \mid When \mid Which \mid How$

$\langle AV \rangle$ ::= $AV_I \mid AV_L \mid AV_D$

$\langle CN \rangle$ ::= $CN_{SWU} \mid CN_{HN} \mid CN_{HLT} \mid CN_{OS} \mid CN_{OP}$

$\langle NN \rangle$ ::= $NN_C \mid NN_O$

$\langle PN \rangle$ ::= $PN_S \mid PN_{HLT} \mid PN_P \mid PN_{HMD} \mid PN_R \mid PN_{HN} \mid PN_{SA} \mid PN_{BN} \mid PN_E \mid PN_{Ent} \mid PN_{BDN} \mid PN_C \mid PN_G \mid PN_{IOG} \mid PN_{PB} \mid PN_{CO}$.

5.3.2 Phase II: Parsing and Tagging

In Phase II, the question is transformed into a pattern of grammatical terms by first parsing the question and then mapping each term to its grammar terminals, as illustrated in Algorithm 2. For sentence such as *"list of movies"* the parsing and mapping is simple since it contains only single words; each word is parsed and mapped individually and will be transformed into the following pattern $[CN_{OS} + P + CN_{OP}]$.

However, for question such as *"What did Alessandro Volta invent in 1800?"* which contains both single and compound words, first compound words will be parsed and extracted then single words, terms will be extracted as follow; *"What"*, *"did"*, *"Alessandro Volta"*, *"invent"*, *"in"*, *"1800"* and the question will be transformed into the following pattern $[QW_{What} + Auxv + PN_C + AV + P + NN_C]$. Some questions or sentences might contain compound words which consist of more than three terms, For example, in a sentence like *"University of Portsmouth Library"* terms will be extracted as follow; *"University of Portsmouth"* will be parsed as one word since it is a compound word and *"Library"* will be parsed as a single word. The following pattern will be formulated $[PN_{IOG} + CN_{OS}]$.

Another example is illustrated in Fig. 5.2 for the question *'What are the symptoms of diabetes?'*. The right-hand side of the figure illustrates the parsing of the question to extract the set of terms using the proposed grammatical rules discussed in Section 5.3.1, while the left-hand side illustrates the mapping of the terms to the grammar non-terminals. As a result of this process, the example question is transformed into the following pattern: $[QW_{What} + LV + D + CN_{OP} + P + CN_{HLT}]$. In this given example the pattern is a representation of the syntactical pattern in level 3 (i.e. the most detailed level).

Algorithm 2 Parsing and Mapping Algorithm

```
Read question q from input file.
Read grammar rules and store it in G.
Parse q and extract the set of terms T
for each t_i in T do
    c_i = Map(t_i, G)                    ▷ This maps term t_i based on G into category c_i
    if c_i is null then
        c_i = PN                         ▷ If no category found for term t_i, assume it is a Proper Noun.
        if c_{i-1} is PN then
            combine(c_{i-1}, c_i)        ▷ Replace any number of consecutive PN with a single PN
        end if
    end if
end for
```

5.3.3 Phase III: Learning and Classification

In this phase, the patterns that were generated in Phase II are used for machine learning, the aim of this phase is to build a model for automatic classification. The classification is done by following the standard process for machine learning, which involves the splitting of the dataset into a training dataset and a test dataset. The training dataset is used for building the model, and the test dataset is used to evaluate the performance of the model. Once a model of satisfactory performance has been identified, it can be used for the classification of unlabelled questions.

5.4 Experimental Study and Results

The objective of the experimental study is to investigate the ability of machine learning classifiers to distinguish between different question types based on the different levels of detail used in the term taxonomy.

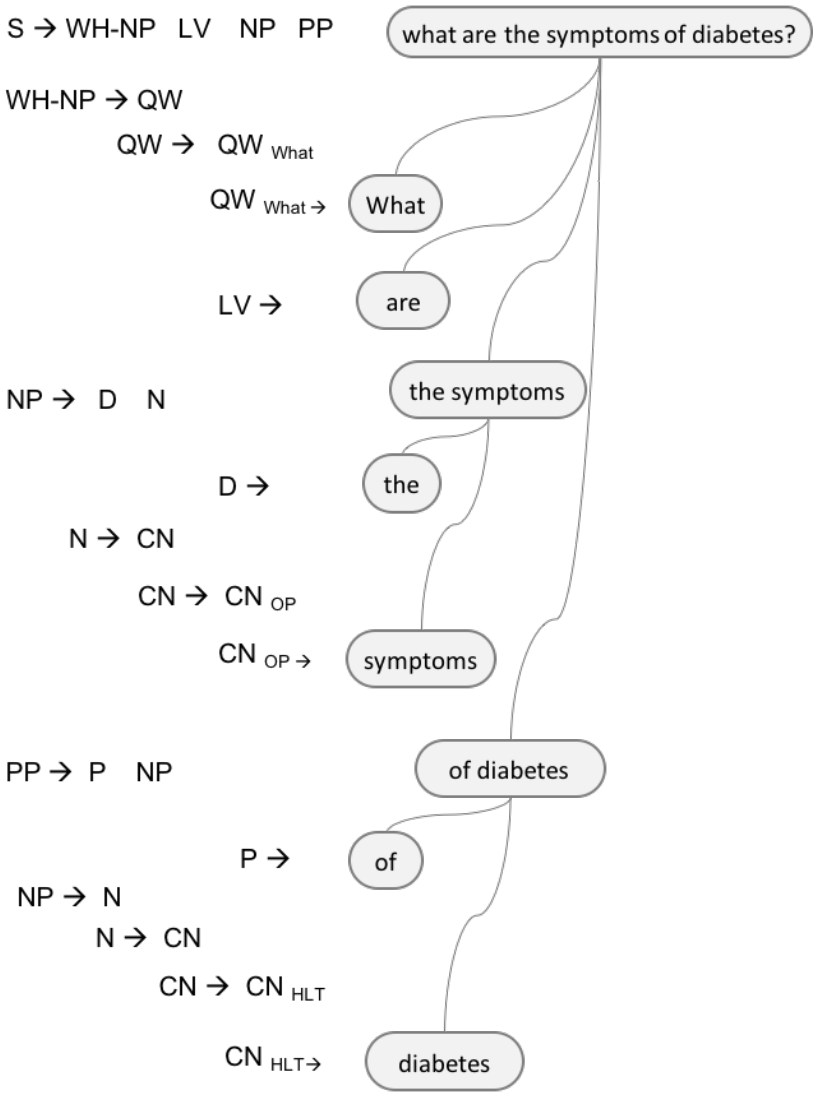

FIGURE 5.2 Phase II: Parsing and Mapping Example

Four machine learning algorithms were used for question classification. Which are; J48, RandomForests (RF), Naive Bayes (NB) and Support Vector Machine (SVM)

A total of 1,160 questions were used that were randomly selected from the datasets that were mentioned in Section 5.2.2: Yahoo Non-Factoid Question, TREC 2007 Question Answering Data and a Wikipedia dataset. Their distribution is given in Table 5.4.

To assess the performance of the machine learning classifiers, the Weka[4] software [37] was used. The experiments were set up using the typical 10-fold cross validation, i.e. the dataset is split into 10 folds, and each fold in used, in turn, for testing, while the other 9 are used for training. The output of the training process is a model, which is then used to classify the questions in the test fold. The labels produced by the model are matched to the true labels and typical performance indicators, such as accuracy, precision, recall, and F-measure, are calculated. The results are presented in the next sub-section.

TABLE 5.4 Data distribution

Question type	Total
Causal	31
Choice	12
Confirmation	321
Factoid	688
Hypothetical	7
List	101

5.4.1 Results

In this section, the results of the machine learning algorithms are presented and analysed for each of the three levels of the term taxonomy.

5.4.1.1 Level-1

Table 5.5 presents classification performance details (Precision, Recall and F-Measure) of the $GQCC_{J48}$, $GQCC_{NB}$, $GQCC_{RF}$ and $GQCC_{SVM}$ classifiers. Results show that Decision Tree ($GQCC_{J48}$) identified correctly (i.e. Recall) 86.6% of the questions, while $GQCC_{SVM}$ identified correctly 85.3% of the questions, $GQCC_{RF}$, 84.7% and $GQCC_{NB}$, 81.6%.

[4] http://www.cs.waikato.ac.nz/ml/weka/

More specifically, looking at where the errors occur, $GQCC_{J48}$ could not identify Causal question and misclassified 3.2% as Confirmation and 96.8% were misclassified as Factoid. For the Choice questions $GQCC_{J48}$ misclassified 41.7% as Confirmation and 25% as Factoid. From the Confirmation questions 1.6% were misclassified as Hypothetical. Furthermore, 0.7% of the Factoid questions were misclassified as Confirmation, 0.15% as Causal, 0.15% as Choice and 1.2% as List. For the List Type of question, 1% were of the questions were misclassified as Confirmation and 87.1% were misclassified as Factoid. Moreover, $GQCC_{J48}$ could not identify Hypothetical questions and incorrectly classify them as Factoid.

$GQCC_{NB}$ classifier incorrectly classified 3.2% of the Causal questions as Choice, 87.1% as Factoid, 3.2% as Confirmation and 6.5% as Hypothetical. In addition, $GQCC_{NB}$ could not identify Choice questions and misclassified 42% as Confirmation, 42% as Factoid and 16% as List. Furthermore, 1.2% of the Confirmation questions were misclassified as Choice, 3.4% as Factoid, 2.1% as Hypothetical and 0.3% as List. For the Factoid questions, 1.1% were misclassified as Causal, 2% as Choice, 1.7% as Confirmation, 2% as Hypothetical and 3% as List. Moreover, 14% of the Hypothetical questions were misclassified as Causal and 57% as Factoid. For the List Type of question $GQCC_{NB}$ incorrectly classified 3% as Confirmation and 86% as Factoid.

Similar to $GQCC_{NB}$ classifier, $GQCC_{RF}$ Classifier could not identify Causal and Choice questions. For the Causal $GQCC_{RF}$ incorrectly classified 3.2% as Confirmation and 96.8% as Factoid. Moreover, 41.7% of the Choice questions were misclassified as Confirmation and 58.3% as Factoid. For the Confirmation questions, 0.3% were misclassified as Choice and 2.8% as Hypothetical. Moreover, 0.6% of the Factoid questions were misclassified as Causal, 0.3% as Choice, 0.9 as Confirmation and 3.2% as List. $GQCC_{RF}$ Could not identify Hypothetical questions misclassified them as Factoid. In addition, 2% of the List questions were misclassified as Confirmation and 80% as Factoid.

Finally, the $GQCC_{SVM}$ classifier could not identify Causal questions and 3.2% of the questions were misclassified as Confirmation and 96.8% were misclassified as Factoid. From the Choice questions 33.3% were misclassified as Confirmation and 33.3% were misclassified as Factoid. Similarly, 1% of the list questions were misclassified as Confirmation and 92% were misclassified as Factoid. Moreover, 2.8% of Confirmation questions were misclassified as Factoid and less than 1% were misclassified as Choice. For the Factoid questions 0.4% were misclassified as Causal, 0.3% were misclassified as Choice, 0.9% were misclassified

as Confirmation and 1.7% were misclassified as List. In addition, most of the Hypothetical questions, i.e. 57%, were misclassified as Factoid.

Comparing the effectiveness of the classifiers in this level, all the classifiers have 0 Precision, Recall, and F-measure for the Causal questions. For the question type Choice, $GQCC_{J48}$ has the highest Precision and F-measure and $GQCC_{SVM}$ has the highest Recall, while the rest of the classifiers have 0 Precision, Recall, and F-measure. Moreover, $GQCC_{J48}$ and $GQCC_{SVM}$ have the highest Precision, and $GQCC_{J48}$ has the highest Recall and F-measure for the Confirmation questions, while $GQCC_{NB}$ has the lowest.

For the Factoid questions, $GQCC_{J48}$ has the highest Precision, Recall, and F-measure, while $GQCC_{NB}$ and $GQCC_{SVM}$ have the lowest Precision and $GQCC_{NB}$ has the lowest Recall and F-measure. Furthermore, $GQCC_{SVM}$ has the highest Precision, Recall, and F-measure for the Hypothetical questions and $GQCC_{J48}$ and $GQCC_{RF}$ have 0 Precision, Recall, and F-measure for this type of question. For the question type List, $GQCC_{J48}$ has the highest Precision and $GQCC_{RF}$ has the highest Recall and F-measure. Furthermore, $GQCC_{NB}$ has the lowest Precision and $GQCC_{SVM}$ has the lowest Recall and F-measure.

TABLE 5.5 Performance of the classifiers in Level (1) - Best results are highlighted in bold, the "*" indicates that the results are significantly better. Precision (P), Recall (R), F-Measure (F).

	$GQCC_{J48}$			$GQCC_{SVM}$			$GQCC_{RF}$			$GQCC_{NB}$		
Accuracy:	**86.6%***			85.3 %			84.7%			81.6%		
Precision:	**0.82**			0.80			0.79			0.78		
Recall:	**0.87**			0.85			0.85			0.82		
F-Measure	**0.83**			0.81			0.81			0.79		
Class:	P	R	F	P	R	F	P	R	F	P	R	F
Causal	0.00	0.00	0.00	0.00	0.00	0.00	0.00 0	0.00	0.00	0.00	0.00	0.00
Choice	**0.80**	**0.33**	**0.47**	0.5	**0.33**	0.40	0.00	0.00	0.00	0.00	0.00	0.00
Conf.	**0.96**	**0.98**	0.97	**0.96**	0.97	0.96	0.96	0.97	0.96	0.93	0.93	0.93
Factoid	**0.84**	**0.98**	**0.90**	0.82	0.97	0.89	0.83	0.95	0.89	0.83	0.92	0.87
Hypo.	0.00	0.00	0.00	**1.00**	**0.43**	**0.60**	0.00	0.00	0.00	0.08	0.29	0.13
List	**0.60**	0.12	0.19	0.37	0.07	0.12	0.45	**0.18**	**0.26**	0.34	0.11	0.17

5.4.1.2 Level-2

Table 5.6 presents classification performance details (Precision, Recall and F-Measure) of the $GQCC_{J48}$, $GQCC_{NB}$, $GQCC_{RF}$ and $GQCC_{SVM}$ classifiers for level 2. Results show that Decision Tree ($GQCC_{J48}$) identified correctly (i.e. Recall) 87.2% of the questions, while

$GQCC_{SVM}$ identified correctly 86.6% of the questions, $GQCC_{RF}$, 85.8% and $GQCC_{NB}$, 81.9%.

More specifically, in this level $GQCC_{J48}$ could not identify Causal and Choice questions and misclassified 3.2% for the Causal questions as Confirmation and 96.8% were misclassified as Factoid. For the Choice questions J48 misclassified 42% as Confirmation and 58% as Factoid. From the Confirmation questions 1.6% were misclassified as Hypothetical and 0.6% as List. Furthermore, 0.9% of the Factoid questions were misclassified as Confirmation, 0.15% as Causal and also 0.15% as List. For the List Type of question, 1% were of the questions were misclassified as Confirmation and 81% were misclassified as Factoid. Moreover, $GQCC_{J48}$ could not identify Hypothetical questions and incorrectly classified them as Factoid.

Similar to $GQCC_{J48}$ classifier, $GQCC_{NB}$ classifier could not identify Causal and Choice questions and incorrectly classified 3.2% of the Causal questions as Confirmation, 90.3% as Factoid, and 6.5% as Hypothetical. For the Choice questions 42% were misclassified as Confirmation and 58% as Factoid. Furthermore, 1.5% of the Confirmation questions were misclassified as Choice, 2.5% as Factoid, 3.1% as Hypothetical and 1.9% as List. For the Factoid questions, 1.9% were misclassified as Causal, 0.3% as Choice, 1.3% as Confirmation, 1.6% as Hypothetical and 2% as List. Moreover, 14% of the Hypothetical questions were misclassified as Causal and 57% as Factoid. For the List Type of question $GQCC_{NB}$ incorrectly classified 5% as Confirmation and 77% as Factoid.

$GQCC_{RF}$ Classifier incorrectly classified 3.5% of the Causal questions as Confirmation and 90.3% as Factoid. Moreover, similar to $GQCC_{NB}$ and $GQCC_{J48}$ classifiers, $GQCC_{RF}$ could not identify Choice questions and misclassified 42% as Confirmation and 58% as Factoid. For the Confirmation questions, 0.6% were misclassified as Choice and 2.5% as Factoid. Moreover, 0.2% of the Factoid questions were misclassified as Choice, 1.2% as Confirmation and 1.6% as List. For the Hypothetical questions $GQCC_{RF}$ misclassified most of them 85.7% as Factoid. In addition, 3% of the List questions were misclassified as Confirmation and 83% as Factoid.

Finally, using $GQCC_{SVM}$, 3.2% of the Causal questions were misclassified as Confirmation and 90.3% were misclassified as Factoid. $GQCC_{SVM}$ is the only classifier in this level to classify Choice questions but misclassified 42% as Confirmation and also 42% as Factoid. Moreover, 1.2% of Confirmation questions were misclassified as Factoid and less

than 1% were misclassified as Choice and List. For the Factoid questions 1.3% were misclassified as Causal, 0.15% were misclassified as Choice and 0.15% as Hypothetical, 0.9% were misclassified as Confirmation and 1.9% were misclassified as List. In addition, 14% of the Hypothetical questions were misclassified as Causal and 43% as Factoid. Finally, 2% of the List questions were misclassified as Confirmation and 71% were misclassified as Factoid.

Comparing the effectiveness of the classifiers in this level, $GQCC_{RF}$ has the highest Precision, while $GQCC_{RF}$ and $GQCC_{SVM}$ both have similar Recall; in addition, $GQCC_{RF}$ has the highest F-measure for the Causal questions, while $GQCC_{NB}$, and $GQCC_{J48}$ have 0 Precision, Recall, and F-measure. For the question type Choice, $GQCC_{SVM}$ has the highest Precision, Recall, and F-measure, while the rest of the classifiers have 0 Precision, Recall, and F-measure. Moreover, $GQCC_{J48}$ has the highest Precision, Recall, and F-measure for the Confirmation questions while $GQCC_{NB}$ has the lowest.

For the Factoid questions, $GQCC_{SVM}$ has the highest Precision and $GQCC_{J48}$ has the highest Recall and F-measure, while $GQCC_{RF}$ has the lowest Precision and $GQCC_{NB}$ has the lowest Recall and F-measure. Furthermore, $GQCC_{RF}$ has the highest Precision and $GQCC_{SVM}$ has the highest Recall and F-measure for the Hypothetical questions; $GQCC_{J48}$ has the lowest Precision, Recall, and F-measure for this type of question. For the question type List, $GQCC_{J48}$ has the highest Precision and $GQCC_{SVM}$ has the highest Recall and F-measure. Furthermore, $GQCC_{NB}$ has the lowest Precision and $GQCC_{RF}$ has the lowest Recall and F-measure.

TABLE 5.6 Performance of the classifiers in Level (2) - Best results are highlighted in bold, the "*" indicates that the results are significantly better. Precision (P), Recall (R), F-Measure (F).

	$GQCC_{J48}$			$GQCC_{SVM}$			$GQCC_{RF}$			$GQCC_{NB}$		
Accuracy:	**87.2% ***			86.6 %			85.8%			81.9%		
Precision:	0.83			**0.84**			0.83			0.79		
Recall:	**0.87**			0.86			0.86			0.82		
F-Measure	0.83			**0.84**			0.82			0.80		
Class:	P	R	F	P	R	F	P	R	F	P	R	F
Causal	0.00	0.00	0.00	0.17	**0.07**	0.09	**1.00**	**0.07**	**0.12**	0.00	0.00	0.00
Choice	0.00	0.00	0.00	**0.40**	**0.17**	**0.24**	0.00	0.00	0.00	0.00	0.00	0.00
Conf.	**0.96**	0.98	**0.97**	0.95	0.97	0.96	0.95	0.97	0.96	0.94	**0.91**	0.92
Factoid	0.84	**0.99**	**0.91**	**0.86**	0.96	0.90	0.83	0.97	0.89	0.84	0.93	0.88
Hypo.	0.00	0.00	0.00	0.75	**0.43**	**0.55**	**1.00**	0.14	0.25	0.08	0.29	0.13
List	**0.86**	0.18	0.29	0.64	**0.27**	**0.38**	0.56	0.14	0.22	0.47	0.18	0.26

5.4.1.3 Level-3

Table 5.7 presents the classification performance details (Precision, Recall and F-Measure) of the $GQCC_{J48}$, $GQCC_{NB}$, $GQCC_{RF}$ and $GQCC_{SVM}$ classifiers for level 3. Results show that Decision Tree ($GQCC_{J48}$) identified correctly (i.e. Recall) 90.1% of the questions, while $GQCC_{SVM}$ identified correctly 88.6% of the questions, $GQCC_{NB}$, 83.5% and $GQCC_{RF}$, 87.7%.

More specifically, looking at where the errors occur, when using $GQCC_{J48}$, 3.2% of the Causal questions were misclassified as Confirmation and 12.9% were misclassified as Factoid. For the Choice questions $GQCC_{J48}$ could not identify this type of question and misclassified 41.1% as Confirmation and 58.3% as Factoid. From the Confirmation questions 0.31% were misclassified as Causal, 0.62% as Factoid and also 0.62% as List. Furthermore, 0.7% of the Factoid questions were misclassified as Confirmation, 1% as Causal, 1% as List and 0.4% as Hypothetical.

For the List Type of question, 1% of the questions were misclassified as Confirmation and 60.4% were misclassified as Factoid. Moreover, $GQCC_{J48}$ could not identify Hypothetical questions and incorrectly classify them as Factoid.

The $GQCC_{NB}$ classifier incorrectly classified 6.5% of the Causal questions as Confirmation, 80.6% as Factoid and 3.2% as List. Similar to $GQCC_{J48}$ classifier, $GQCC_{NB}$ could not identify Choice questions and misclassified 41.7% as Confirmation and 58.3% as Factoid. Furthermore, 0.9% of the Confirmation questions were misclassified as Choice, 3.4% as Factoid, 2% as Hypothetical and 0.9% as List. For the Factoid questions, 1.3% were misclassified as Causal, 0.43% as Choice, 2.5% as Confirmation, 0.87% as Hypothetical and 2.2% as List. Moreover, 14.3% of the Hypothetical questions were misclassified as Causal and 57.1% as Factoid. For the List Type of question $GQCC_{NB}$ incorrectly classified 7% as Confirmation and 65.3% as Factoid.

The $GQCC_{RF}$ classifier incorrectly classified 6.4% of the Causal questions as Confirmation and 58.3% as Factoid. Similar to $GQCC_{J48}$ and $GQCC_{NB}$ classifiers, $GQCC_{RF}$ could not identify Choice questions and misclassified 41.7% as Confirmation and 58.3% as Factoid. For the Confirmation questions, 0.6% were misclassified as Choice and 3.4% as Factoid. Moreover, 1.2% of the Factoid questions were misclassified as Confirmation and 0.7% as List. Hypothetical questions were 71.4% misclassified as Factoid. In addition, 2%

of the List questions were misclassified as Confirmation and 72.3% as Factoid.

Finally, using $GQCC_{SVM}$, 3.2% of the Causal questions were misclassified as Confirmation and 32.2% were misclassified as Factoid. From the Choice questions 41.7% were misclassified as Confirmation and 33.3% were misclassified as Factoid. Similarly, 4% of the List questions were misclassified as Confirmation and 45.5% were misclassified as Factoid. These results indicate that $GQCC_{SVM}$ could not distinguish between Causal, Choice and List types of questions and incorrectly classified most of them as Confirmation and Factoid questions. Moreover, 1.6% of Confirmation questions were misclassified as Factoid and less than 1% were misclassified as Choice or List. For the Factoid questions 4.6% were misclassified as List, 1.2% were misclassified as Causal, 1% were misclassified as Confirmation and less than 1% were misclassified as Choice. In addition, most of the Hypothetical questions 57.1% were misclassified as Factoid.

Comparing the effectiveness of the classifiers, $GQCC_{RF}$ has the highest Precision for the Causal questions and $GQCC_{J48}$ has the highest Recall and F-measure, while $GQCC_{NB}$ has the lowest Precision, Recall, and F-measure. For the question type Choice, $GQCC_{SVM}$ has the highest Precision, Recall, and F-measure, while the rest of the classifiers have 0 Precision, Recall, and F-measure. Moreover, $GQCC_{J48}$ has the highest Precision, Recall, and F-measure for the Confirmation questions, while $GQCC_{NB}$ has the lowest.

For the Factoid questions, $GQCC_{SVM}$, $GQCC_{RF}$, and $GQCC_{J48}$ have the highest Precision, Recall, and F-measure respectively, while $GQCC_{NB}$ and $GQCC_{RF}$ have the lowest Precision and $GQCC_{NB}$ has the lowest Recall and F-measure. Furthermore, $GQCC_{RF}$ and $GQCC_{SVM}$ have the highest Precision for the Hypothetical questions and $GQCC_{SVM}$ has the highest Recall and F-measure. $GQCC_{J48}$ has the lowest Precision, Recall, and F-measure for this type of question. For question type List $GQCC_{RF}$ has the highest Precision and $GQCC_{SVM}$ has the highest Recall and F-measure. Furthermore, $GQCC_{SVM}$ has the lowest Precision and $GQCC_{RF}$ has the lowest Recall, while $GQCC_{NB}$ has the lowest F-measure.

The results show (Figure 5.3) that with each level there is an improvement in the results when moving from *level 1* to *level 2* and from *level 2* to *level 3*. The improvement in the performance from *level 1* to *level 2* is marginal and there is an increase in the performance from *level 2* to *level 3*. In addition, the results indicate that $GQCC_{J48}$ significantly preformed better than $GQCC_{SVM}$, $GQCC_{RF}$ and $GQCC_{NB}$ in all three levels. Weka corrected t-test were used with the threshold value of 0.05 (i.e. p-value $<$0.05).

TABLE 5.7 Performance of the classifiers in Level (3) - Best results are highlighted in bold, the "*" indicates that the results are significantly better. Precision (P), Recall (R), F-Measure (F).

	$GQCC_{J48}$			$GQCC_{SVM}$			$GQCC_{RF}$			$GQCC_{NB}$		
Accuracy:	**90.1 %***			88.6%			87.7%			83.5%		
Precision:	**0.89**			0.88			0.87			0.81		
Recall:	**0.90**			0.89			0.88			0.84		
F-Measure:	**0.89**			0.88			0.85			0.82		
Class:	P	R	F	P	R	F	P	R	F	P	R	F
Causal	0.72	**0.84**	**0.78**	0.71	0.65	0.68	**1.00**	0.19	0.32	0.23	0.09	0.14
Choice	0.00	0.00	0.00	**0.43**	**0.25**	**0.32**	0.00	0.00	0.00	0.00	0.00	0.00
Conf.	**0.96**	**0.98**	**0.97**	0.95	0.97	0.96	0.95	0.96	0.95	0.91	0.93	0.92
Factoid	0.89	0.97	**0.93**	**0.90**	0.93	0.92	0.85	**0.98**	0.91	0.85	0.93	0.89
Hypo.	0.00	0.00	0.00	**1.00**	**0.43**	**0.60**	**1.00**	0.29	0.44	0.13	0.29	0.18
List	0.81	0.39	0.52	0.60	**0.51**	**0.55**	**0.84**	0.26	0.39	0.61	0.28	0.38

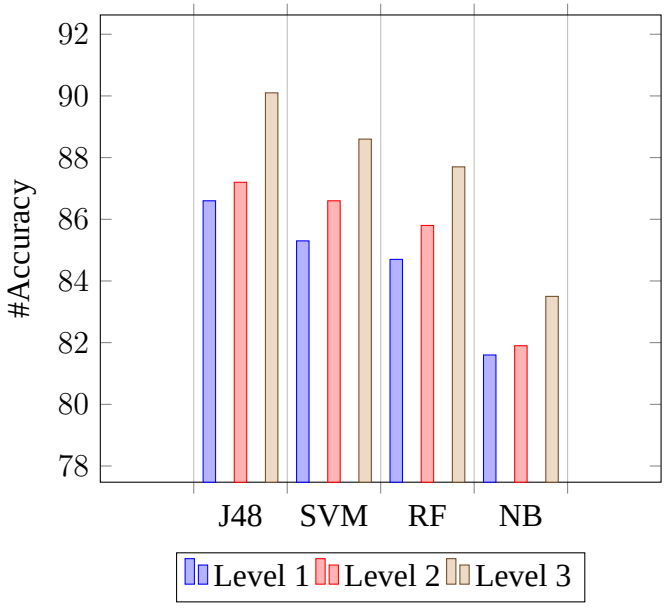

FIGURE 5.3 Accuracy of the classifiers in level 1, 2 and 3

Level 1 and level 2 contain general grammatical categories of the English language. When only the higher level categories are used (i.e. level 1), while there are variations between the different learning algorithms, the overall picture is that the best performance occurs for Confirmation and Factoid questions and the worst performance for Causal questions. In this level, all of the classifiers could not identify at least one question type. $GQCC_{SVM}$ could not identify Causal questions, RF could not identify Causal, Choice and Hypothetical questions, $GQCC_{J48}$ could not identify Causal and Hypothetical questions and $GQCC_{NB}$ could not identify Causal and Choice questions.

When sub-categories of the English main syntactic categories are used, i.e. level 2, a dramatic improvement could be noticed in the performance of all classifiers. $GQCC_{SVM}$ is the only classifier that could identify all type of questions but similar to the performance in level 1 $GQCC_{NB}$, $GQCC_{J48}$ and $GQCC_{RF}$ could not identify at least one question type. $GQCC_{RF}$ could not identify Choice questions, but for the Causal and Hypothetical questions $GQCC_{RF}$ has a recall of 1 for these classes which indicates that there are no false positives, i.e. all instances identifies by the models as Causal or Hypothetical are truly these two types. Furthermore, $GQCC_{J48}$ could not identify Choice and Hypothetical questions in addition to Causal, in contrast to level 1 in which $GQCC_{J48}$ was able to classify Choice questions. $GQCC_{NB}$ also could not identify Causal and Choice questions. The sub-categories at level 2 have also marginally improved the performance for the some question types like List for the classifiers $GQCC_{SVM}$, $GQCC_{J48}$ and $GQCC_{NB}$, Factoid for $GQCC_{RF}$ and $GQCC_{J48}$, Hypothetical for $GQCC_{RF}$, Causal for the classifiers $GQCC_{SVM}$ and $GQCC_{RF}$ and finally Confirmation for the $GQCC_{SVM}$ classifier.

Level 3, which includes the domain-specific grammatical categories, led to significant improvements of the performance of all classifiers. In this level $GQCC_{J48}$ and $GQCC_{NB}$ could identify Causal questions. Regarding Choice question, $GQCC_{RF}$, $GQCC_{J48}$ and $GQCC_{NB}$ still could not identify this type of question; in addition, similar to level 1 and 2, $GQCC_{J48}$ with the more detailed grammar could not identify Hypothetical questions. $GQCC_{SVM}$ is the only classifier that could identify and classify all type of questions.

These results indicate that the syntactic categories related to different domain-specific types of Common Nouns, Numeral Numbers and Proper Nouns enable the machine learning algorithms to better differentiate between different question types.

The Third objective was to investigate the optimal level of detail for the domain-related

syntactic categories. The results from the experiments indicate that the answer to this question is that the highest level of detail leads to the best classification performance. The structure with the 3 levels is more useful for an automatic approach to question identification, facilitating a faster mapping process. The Fourth objective was about which machine learning algorithms are best suited to classification of question types, when using the data representation proposed in the GQCC framework. Naive Bayes ($GQCC_{NB}$), which is known to perform well on textual data, leads to the lowest performance models in the experiments (but not by much), while J48 ($GQCC_{J48}$) leads to the best performing model. When using the domain-specific syntactic categories (level 3) 5.7, $GQCC_{RF}$ and $GQCC_{SVM}$ were very close in performance. Consequently, the consistent performance of the classifiers validates the contribution of the new representation, with its domain-specific information and preservation of order, to the high classification performance.

The Fifth objective was about the classification performance of the proposed approach (GQCC) in comparison with state-of-the-art approaches. This is discussed in detail in the following section.

5.4.2 Performance comparison with other question classification approaches

For the objective of validating the proposed approach (GQCC) in improving the classification accuracy and the identification of different type of questions and to compare the classification performance of the proposed approach with the state-of-the-art approaches, experiments have been conducted using features classifier model based on the most used features in previous works such as n-gram in which $n = 2$, Bag-of-Words, Snowball Stemmer and stop words remover.

Similar to previous experiments in section 4.4, to assess the performance of the machine learning classifier the experiments were set up using the typical 10-fold cross validation, i.e. the dataset is split into 10 folds, and each fold in used, in turn, for testing, while the other 9 are used for training. The output of the training process is a model, which is then used for classification in the test fold. The labels produced by the model are matched to the true labels and typical performance indicators, such as Accuracy, Precision, Recall, and F-Measure, are calculated. In addition, Four machine learning algorithms, were used for question classification. Which are; J48, Random forests (RF), Naive Bayes (NB) and Support Vector Machine

(SVM).

5.4.2.1 Results

Table 5.8 presents classification performance details (Precision, Recall and F-Measure) of the n-gram$_{J48}$, n-gram$_{NB}$, n-gram$_{RF}$ and n-gram$_{SVM}$ classifiers using broder's extended query categories. Results show that Decision Tree (n-gram$_{J48}$) identified correctly (i.e. Recall) 81.1% of the queries, while n-gram$_{SVM}$ identified correctly 71.2% of the queries, n-gram$_{RF}$, 77.1% and n-gram$_{NB}$, 68.4%.

When using features such as n-gram, Bag-of Words, Snowball Stemmer and stop word remover most of the classifiers had low Precision, Recall, and F-Measure. All the classifiers had 0 Precision, Recall, and F-Measure for Hypothetical questions while n-gram$_{J48}$ and n-gram$_{RF}$ had 0 Precision, Recall, and F-Measure for Choice questions and n-gram$_{NB}$ could not identify List and Causal questions.

These results validate that using domain-specific information improves the classification accuracy and could be used in the identification of different type of questions in addition to domain categories.

TABLE 5.8 Performance of the classifiers using the features and n-gram framework - $GQCC_{J48}$ results are highlighted in bold. Precision (P), Recall (R), F-Measure (F).

	$GQCC_{J48}$			n-gram$_{J48}$			n-gram$_{SVM}$			n-gram$_{RF}$			n-gram$_{NB}$		
Accuracy:	**90.1%**			81.1%			71.2%			77.1%			68.4%		
Class:	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
Causal	**0.72**	**0.84**	**0.78**	0.05	0.03	0.04	0.20	0.03	0.06	1.00	0.03	0.07	0.00	0.00	0.00
Choice	**0.00**	**0.00**	**0.00**	0.00	0.00	0.00	0.50	0.18	0.27	0.00	0.00	0.00	0.00	0.00	0.00
Conf.	**0.96**	**0.98**	**0.97**	0.92	0.93	0.92	0.69	0.71	0.70	0.80	0.74	0.76	0.81	0.39	0.52
Factoid	**0.89**	**0.97**	**0.93**	0.83	0.91	0.87	0.76	0.83	0.79	0.77	0.94	0.85	0.67	0.96	0.79
Hypo.	**0.00**	**0.00**	**0.00**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
List	**0.81**	**0.39**	**0.52**	0.21	0.12	0.15	0.28	0.20	0.23	0.17	0.01	0.02	0.00	0.00	0.00

5.4.3 Dealing with class imbalance

For the objective of evaluating the impact of handling class imbalance in the classification accuracy, experiments have been conducted using the SMOTE algorithm. The Synthetic Minority Over-sampling Technique (SMOTE) [18] is one of the most popularly used sampling technique to handle imbalance data. SMOTE over-samples instances of the minority (abnormal) class which helps for achieving better classifier performance. After testing different machine learning classifiers such as J48, SVM, NB and RF; Naive Bayes was used

as the machine learning algorithm for the classification, as it preformed better with SMOTE algorithm than the other classifiers.

To show the effectiveness of handling imbalance data on the classification performance, two experiments were conduct (1) using NB ($GQCC_{NB}$) without applying SMOTE algorithm and (2) using NB ($GQCC_{NB_{SMOTE}}$) with the implementation of SMOTE algorithm. Similar to previous experiments the 1,160 questions were used from the three datasets (1) TREC 2007 Question Answering Data, (2) a Wikipedia dataset and (3) Yahoo Non-Factoid Question Dataset.

5.4.3.1 Results

Table 5.9 presents classification performance details (Precision, Recall and F-Measure) of the $GQCC_{NB}$ classifier and the performance details of the $GQCC_{NB_{SMOTE}}$ classifier with the use of the SMOTE algorithm. The results indicate that when handling imbalance classes the performance of the classifier is improved, as shown in Table 5.4. Choice, Causal and Hypothetical questions have much fewer instances, and without applying the SMOTE algorithm the classifier had poor performance especially with these three classes. However, when the SMOTE algorithm is applied, the performance of the classifier has been improved and the overall accuracy has increased.

TABLE 5.9 NB classifier performance without/with the implementation of SMOTE algorithm

	$GQCC_{NB}$			$GQCC_{NB_{SMOTE}}$		
Question Types	Precision	Recall	F-Measure	Precision	Recall	F-Measure
Causal	**0.231**	**0.097**	**0.136**	**0.621**	**0.581**	**0.600**
Choice	**0.000**	**0.000**	**0.000**	**0.154**	**0.167**	**0.160**
Confirmation	0.906	0.928	0.917	0.944	0.941	0.942
Factoid	0.850	0.927	0.887	0.870	0.955	0.911
Hypothetical	**0.133**	**0.286**	**0.182**	**0.417**	**0.714**	**0.526**
List	0.609	0.277	0.381	0.613	0.188	0.288
Overall	0.814	0.835	0.818	0.851	0.865	0.847

Furthermore, these results show that $GQCC_{NB}$ is effective in the identification and classification of Confirmation and Factoid questions. In addition, $GQCC_{NB}$ could not distinguish between Causal, Choice, Hypothetical and List types of questions and incorrectly classified most of them as Confirmation and Factoid questions. However, when applying SMOTE algorithm classification of most question types and the performance has been improved. For

example, when the SMOTE algorithm is not applied, $GQCC_{NB}$ could correctly classified (Recall) less than 1% of the Causal questions, and could not identify any of the Choice questions. Furthermore, $GQCC_{NB}$ classified correctly 92.8% of the Confirmation questions and 92.7% of the Factoid questions. In addition, 28.6% of the Hypothetical questions were correctly classified while the classification accuracy of the List questions were 27.7%.

On the contrary, when SMOTE algorithm is applied, $GQCC_{NB_{SMOTE}}$ correctly classified 58.1% of the Causal questions and 16.7% of the Choice questions. In addition, classification of Factoid, Confirmation and Hypothetical questions achieved a higher recall when handling imbalance classes, i.e. 95.5%, 94.1% and 71.4% accuracy respectively. Moreover, classification of List questions shows a lower recall (18.8%) with the implementation of SMOTE but higher precision. Overall, the results validate that when handling the problem of imbalanced classes, the performance improves and classification accuracy increases.

5.5 Discussion

In this section, the performance of the previous methods is discussed in terms of accuracy. [73] proposed a hierarchical classifier that classifies questions into fine grained classes, using Sparse Network of Winnows (SNoW); the proposed approach achieved an accuracy of 92.5% for coarse grained classes and 85% for fine grained classes when using only syntactical features; after adding semantic features the accuracy accounted for 89.3%.

Most previous works were based on Li and Roth classification of question and deal with factoid questions only. [159] used bag-of-words features on different machine learning algorithms; SVM performed better compared with the other classifiers like KNN and NB. SVM achieved an accuracy of 80.2% with fine grained classes and 85.8% with coarse grained classes.

[48] used head word features and wordNet in addition to unigrams; their liner SVM and maximum entropy models reach the accuracy of 89.2% and 89% respectively. The statistical classifier in [87] is based on SVM and achieved an accuracy of 90.2% using coarse grained classes and 83.6% using fine grained classes. [70] classified factoid questions using head Noun tagging combining syntactical and semantic features; they uses Conditional Random Fields (CRFs) and SVM; the model achieved an accuracy of 85.6%. [102] proposed an approach which is based on question patterns and designed features; they achieved an accuracy of 95.2%

and 91.6% for coarse grained classes and fine grained classes respectively using SVM.

Even-though these approaches achieved good accuracy rate, they have used Li and Roth classification of questions, which is based on a large number of categories. In addition, this classification only focuses on solving the problem of classifying and identifying factoid types of question. Furthermore, the majority of these previous works used SVM for the classification process; in this research, experiments have shown that other classifiers like J48 could be used for the classification with good results.

Furthermore, [14] classified open ended questions using SVM and achieved an accuracy of 74.6% on average. However, the data in this work were collected from textbook and references, which are not representative of questions typically asked in question-answering systems. In addition, some of the data attributes have been removed like stop words,"s" for plural words and "ly" for adverbs, which are important to identify question types. For example, plural terms are one of the main identification feature of question type "List".

In conclusion, The proposed approach (GQCC) outperforms the previous ones due to the ability of this approach to classify different questions types, not just Factoid. In addition, this approach uses domain-specific information which facilitate the identification of domain categories, unlike previous works which focus only on the type of question.

5.6 Summary of Chapter

In this chapter, the Customizable Grammar Framework (CGF) for user intent text classification was applied to question classification problem in which a Grammar Based Framework for question categorization and classification (GQCC) was proposed with the objective of creating a question categorization and classification framework that could easily be applied to different question-answering systems by creating domain specific grammatical rules and patterns for each type of question. In addition, general and domain-specific syntactic categories were identified and different types of question were introduced and analysed. Furthermore, the results showed that the proposed solution led to a good performance in classifying questions and outperforms the previous ones due to the ability of this approach to classify different type of questions and not just factoid.

CHAPTER 6

Summary and Future Work

This thesis presents a grammar-based framework for user-intent text classification. The proposed framework addresses the problem of text identification and classification in different domains. In this chapter, the contribution is summarised in section 6.1 and directions for future work in this area of research are presented in Section 6.2.

6.1 Summary of Contributions

This thesis proposes a grammar-based text classification framework; Customizable Grammar Framework (CGF) for the automatic classification of text through machine learning; the proposed framework takes advantage of domain-specific information and preserve the structure of the text.

A new representation was proposed, in which text is represented as a syntactical pattern, i.e. a pattern formed of grammatical categories corresponding to the terms in the text. To transform the text into this representation, a formal grammar-based approach was proposed. The proposed approach addressed one of the major issues in text representation, i.e. large sparse datasets, by requiring a significantly smaller number of features. The framework consists of three main phases: (1) grammar; (2) parsing and tagging; (3) learning and classification. In addition, it has been applied to the query classification problem in search engines and question classification problem in question answering systems.

For the query classification, the Grammar Based Framework for Query Classification (GQC) was proposed to automatically classify queries through machine learning. In addition, an analysis of web search queries was provided by identifying the grammatical structure of each type of search query. Furthermore, to investigate the ability of the machine learning classifiers to distinguish between different query types based on the different levels of detail

used in the term taxonomy, four machine learning algorithms, were used for query classification. Results indicated that the proposed approach outperformed previous ones, both overall, as well as for each type of query. Furthermore, for the objective of validating the proposed approach in improving the classification accuracy and the identification of the different type of queries, additional experiments have been conducted using a classifier model which consists of features such as n-gram, Bag-of-Words, Snowball Stemmer and stop word remover. The final results have validated that using domain-specific information and preserving the structure of the query improve the classification accuracy and the identification of different types of queries.

For the question classification, the Grammar Based Framework for question categorization and classification (GQCC) was proposed to automatically classify questions through machine learning. In addition, a new question categorization was proposed which is based on the general English question types and the simple type of questions that are asked by most people. Furthermore, to investigate the ability of machine learning classifiers to distinguish between different question types based on the different levels of detail used in the term taxonomy, four machine learning algorithms, were used for question classification. Results showed that the proposed approach outperformed the previous ones due to the ability of the proposed approach to classify different questions types, not just factoid. In addition, the proposed approach used domain-specific information which facilitated the identification of domain categories, unlike previous works which focus only on the type of question.

Similar to query classification, additional experiments have been conducted using a classifier model which consists of features such as n-gram, Bag-of-Words, Snowball Stemmer and stop word remover; for the objective of validating the proposed approach in improving the classification accuracy and the identification of the different type of question. The final results have validated that using domain-specific information and preserving the structure of the question improve the classification accuracy and the identification of different types of questions in addition to domain categories. Furthermore, to evaluate the impact of handling class imbalance on the classification accuracy, experiments have been conducted using the SMOTE algorithm. The results showed that handling class imbalance led to a good performance in classifying questions.

These results and findings showed that the Customizable Grammar Framework (CGF) for user-intent text classification could be applied to other text classification problems in different

domains.

To summarise, the contributions are:

1) A Customizable Grammar Framework (CGF) for user-intent text classification which addressed the limitations of general approaches in text classification by taking into account the structure of the text. In addition, it combines domain knowledge with a formal grammar by the use of grammatical rules and patterns. CGF has been applied to two different information retrieval applications, which are search engines for the classification of queries and questions answering systems for the classification and categorization of questions and in both domains CGF has achieved a high level of accuracy.

2) Grammatical rules and patterns which helped in the enhancement of the problem of terms ambiguity and the identification of different terms.

3) A syntax-based parsing and tagging which is used to assign not just general PoS tags but also domain specific which helped in the categorization and classification of text in different domains such as search engines and question answering systems. This includes a tag-set which consists of 10,440 different words which have been labelled to general and domain specific PoS categories.

4) A Grammar-Based Framework for Query Classification (GQC) which helped in improving query classification and the identification of different users' intents. This is done by (1) the analysis of query grammatical structure and characteristics, (2) developing domain specific terms categories and (3) creating domain specific grammatical rules and search type syntactical patterns. Results showed that this approach outperforms previous ones in terms of classification performance in which GQC using RF ($GQCC_{RF}$) classifier has outperformed other classification methods with 99.6% accuracy.

5) A Grammar-Based Framework for Question Categorization and Classification (GQCC) which helped in improving question classification and identification. This is done by (1) the analysis of the questions structure and characteristics, (2) introducing a new questions categorization, (3) developing domain specific terms categories and (4) creating domain specific grammatical rules and patterns. Results indicated that using syntactic

categories related to different domain-specific types enable the machine learning algorithms to better differentiate between different question types in which GQCC using J48 ($GQCC_{J48}$) classifier has outperformed other classification methods with 90.1% accuracy.

6.2 Directions for Future Work

The Customizable Grammar Framework (CGF) framework and this research can be further advanced by exploring the following areas:

- Investigate the effect of adding more domain specific PoS categories from different sources on the parsing and tagging phase and how this will effect the classification accuracy and the ability of the framework to be applied to other applications and domains, in which CGF domain specific PoS tags have been developed from different queries/questions datasets only.
- Evaluate the impact of using more detailed grammatical rules, syntactic categories and domain-specific information on the classification, as CGF deals with a simple version of the English grammar which was tailored to deal with real-world text classification problem.
- Investigate the impact of using different types of English grammar (e.g Systemic Functional Grammar (SFG) on the identification classification process, as CGF is based on the context-free grammar in the Backus Normal Form (BNF).
- Develop an automatic framework/method for designing and generating the grammar through analyzing the text and domain knowledge.

Furthermore, the promising results of this research opens up the opportunities for many other interesting research directions including:

- *Text Identification and Classification:* text in other domains with similar classification problems such as the identification and classification of fake news through knowledge learning and the identification of patterns and structures. This could be done by using the customizable Grammar Framework (CGF) for user-intent text classification through using domain knowledge, grammatical rules and patterns.

- *Parsing and Tagging:* one of the fundamental phases in text processing is parsing and tagging and most of the existing tools are based on generic NLP tags which do not capture domain-related information. Developing a customizable domain specific parsing and tagging tool will help in domain such as web queries which usually do not preserve the formal English grammar like word order, and no labeled syntactic trees for such domain are available.

- *Multi-Labels classification and categorization:* when dealing with real-world text classification in domains such as search engines and question answering systems many queries/questions may have more than one label and most previous approaches excluded such cases as most machine learning algorithms based framework for query/question are design and built to classify single-labels text. Developing a Multi-Labels text classification and categorization framework would be useful.

- *Class Imbalance:* in which the classification of imbalanced data has been a key problem in machine learning and data mining and many information retrieval applications such as question answering systems may suffer from the problem of class imbalance. This problem affects the classification results and accuracy, so applying different imbalance algorithms e.g (cost-sensitive and SMOTE) may lead to the improvement and a good performance in classifying questions.

References

[1] Altrabsheh, N., Cocea, M., Fallahkhair, S.: Sentiment analysis: towards a tool for analysing real-time students feedback. In: Tools with Artificial Intelligence (ICTAI), 2014 IEEE 26th International Conference on. pp. 419–423. IEEE (2014)

[2] Ashkan, A., Clarke, C.L., Agichtein, E., Guo, Q.: Classifying and characterizing query intent. In: European Conference on Information Retrieval. pp. 578–586. Springer (2009)

[3] Baeza-Yates, R., Calderón-Benavides, L., González-Caro, C.: The intention behind web queries. In: International Symposium on String Processing and Information Retrieval. pp. 98–109. Springer (2006)

[4] Barr, C., Jones, R., Regelson, M.: The linguistic structure of english web-search queries. In: Proceedings of the conference on empirical methods in natural language processing. pp. 1021–1030. Association for Computational Linguistics (2008)

[5] Basu, T., Murthy, C.: Effective text classification by a supervised feature selection approach. In: 2012 IEEE 12th International Conference on Data Mining Workshops. pp. 918–925. IEEE (2012)

[6] Bauer, A., Braun, M., Müller, K.R.: Accurate maximum-margin training for parsing with context-free grammars. IEEE transactions on neural networks and learning systems 28(1), 44–56 (2017)

[7] Beitzel, S.M., Jensen, E.C., Frieder, O., Grossman, D., Lewis, D.D., Chowdhury, A., Kolcz, A.: Automatic web query classification using labeled and unlabeled training

data. In: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 581–582. ACM (2005)

[8] Benamara, F.: Cooperative question answering in restricted domains: the webcoop experiment (2004)

[9] Bhatia, S., Brunk, C., Mitra, P.: Analysis and automatic classification of web search queries for diversification requirements. Proceedings of the American Society for Information Science and Technology 49(1), 1–10 (2012)

[10] Brants, T.: Tnt: a statistical part-of-speech tagger. In: Proceedings of the sixth conference on Applied natural language processing. pp. 224–231. Association for Computational Linguistics (2000)

[11] Breiman, L.: Random forests. Machine learning 45(1), 5–32 (2001)

[12] Broder, A.: A taxonomy of web search. In: ACM Sigir forum. vol. 36, pp. 3–10. ACM (2002)

[13] Bu, F., Zhu, X., Hao, Y., Zhu, X.: Function-based question classification for general qa. In: Proceedings of the 2010 conference on empirical methods in natural language processing. pp. 1119–1128. Association for Computational Linguistics (2010)

[14] Bullington, J., Endres, I., Rahman, M.: Open ended question classification using support vector machines. MAICS 2007 (2007)

[15] Calderón-Benavides, L., González-Caro, C., Baeza-Yates, R.: Towards a deeper understanding of the user's query intent. In: SIGIR 2010 Workshop on Query Representation and Understanding. pp. 21–24 (2010)

[16] Carterette, B., Pavlu, V., Fang, H., Kanoulas, E.: Million query track 2009 overview. In: TREC (2009)

[17] Charniak, E.: A maximum-entropy-inspired parser. In: Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference. pp. 132–139. Association for Computational Linguistics (2000)

[18] Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research 16, 321–357 (2002)

[19] Chen, D., Manning, C.: A fast and accurate dependency parser using neural networks. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 740–750 (2014)

[20] Chomsky, N.: Lectures on government and binding: The Pisa lectures. No. 9, Walter de Gruyter (1993)

[21] Cohen, W.W.: Fast effective rule induction. In: Proceedings of the twelfth international conference on machine learning. pp. 115–123 (1995)

[22] Conneau, A., Schwenk, H., Barrault, L., Lecun, Y.: Very deep convolutional networks for text classification. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers. vol. 1, pp. 1107–1116 (2017)

[23] Cortes, C., Vapnik, V.: Support-vector networks. Machine learning 20(3), 273–297 (1995)

[24] De Marneffe, M.C., MacCartney, B., Manning, C.D., et al.: Generating typed dependency parses from phrase structure parses. In: Proceedings of LREC. vol. 6, pp. 449–454. Genoa Italy (2006)

[25] Figueroa, A.: Exploring effective features for recognizing the user intent behind web queries. Computers in Industry 68, 162–169 (2015)

[26] Fürnkranz, J., Widmer, G.: Incremental reduced error pruning. In: Proceedings of the 11th International Conference on Machine Learning (ML-94). pp. 70–77 (1994)

[27] Ganchev, K., Hall, K., McDonald, R., Petrov, S.: Using search-logs to improve query tagging. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2. pp. 238–242. Association for Computational Linguistics (2012)

[28] Gildea, D.: Corpus variation and parser performance. In: Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing. pp. 167–202 (2001)

[29] Giménez, J., Marquez, L.: Fast and accurate part-of-speech tagging: The svm approach revisited. Recent Advances in Natural Language Processing III pp. 153–162 (2004)

[30] Giménez, J., Marquez, L.: Svmtool: A general pos tagger generator based on support vector machines. In: In Proceedings of the 4th International Conference on Language Resources and Evaluation. Citeseer (2004)

[31] Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J., Smith, N.A.: 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. Association for Computational Linguistics (2011)

[32] Glorot, X., Bordes, A., Bengio, Y.: Domain adaptation for large-scale sentiment classification: A deep learning approach. In: Proceedings of the 28th international conference on machine learning (ICML-11). pp. 513–520 (2011)

[33] Gong, Z., Yu, T.: Chinese web text classification system model based on naive bayes. In: E-Product E-Service and E-Entertainment (ICEEE), 2010 International Conference on. pp. 1–4. IEEE (2010)

[34] González-Caro, C., Baeza-Yates, R.: A Multi-faceted Approach to Query Intent Classification, pp. 368–379. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)

[35] Greenbaum, S., Nelson, G.: An introduction to English grammar. Pearson Education (2002)

[36] Hacioglu, K., Ward, W.: Question classification with support vector machines and error correcting codes. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: companion volume of the Proceedings of HLT-NAACL 2003–short papers-Volume 2. pp. 28–30. Association for Computational Linguistics (2003)

[37] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. ACM SIGKDD explorations newsletter 11(1), 10–18 (2009)

[38] Han, H.q., Zhu, D.H., Wang, X.f.: Semi-supervised text classification from unlabeled documents using class associated words. In: Computers & Industrial Engineering, 2009. CIE 2009. International Conference on. pp. 1255–1260. IEEE (2009)

[39] Hao, T., Xie, W., Wu, Q., Weng, H., Qu, Y.: Leveraging question target word features through semantic relation expansion for answer type classification. Knowledge-Based Systems 133, 43–52 (2017)

[40] Hao, T., Xie, W., Xu, F.: A wordnet expansion-based approach for question targets identification and classification. In: Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data, pp. 333–344. Springer (2015)

[41] Hardy, H., Cheah, Y.N.: Question classification using extreme learning machine on semantic features. Journal of ICT Research and Applications 7(1), 36–58 (2013)

[42] Hasan, A.M., Zakaria, L.Q.: Question classification using support vector machine and pattern matching. Journal of Theoretical and Applied Information Technology 87(2), 259 (2016)

[43] Hernández, I., Gupta, P., Rosso, P., Rocha, M.: A simple model for classifying web queries by user intent. In: 2nd Spanish Conference on Information Retrieval, CERI-2012. pp. 235–240 (2012)

[44] Herrera, M.R., de Moura, E.S., Cristo, M., Silva, T.P., da Silva, A.S.: Exploring features for the automatic identification of user goals in web search. Information Processing & Management 46(2), 131 – 142 (2010), http://www.sciencedirect.com/science/article/pii/S0306457309001058

[45] Higashinaka, R., Isozaki, H.: Corpus-based question answering for why-questions. (2008)

[46] Ho, T.K.: Random decision forests. In: Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on. vol. 1, pp. 278–282. IEEE (1995)

[47] Højgaard, C., Sejr, J., Cheong, Y.G.: Query categorization from web search logs using machine learning algorithms. International Journal of Database Theory and Application 9(9), 139–148 (2016)

[48] Huang, Z., Thint, M., Qin, Z.: Question classification using head words and their hypernyms. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. pp. 927–936. Association for Computational Linguistics (2008)

[49] Jansen, B.J., Booth, D.L., Spink, A.: Determining the informational, navigational, and transactional intent of web queries. Information Processing & Management 44(3), 1251–1266 (2008)

[50] Jia, R., Liang, P.: Data recombination for neural semantic parsing. arXiv preprint arXiv:1606.03622 (2016)

[51] Joachims, T.: Text categorization with support vector machines: Learning with many relevant features. Machine learning: ECML-98 pp. 137–142 (1998)

[52] Kathuria, A., Jansen, B.J., Hafernik, C., Spink, A.: Classifying the user intent of web queries using k-means clustering. Internet Research 20(5), 563–581 (2010)

[53] Kellar, M., Watters, C., Shepherd, M.: A goal-based classification of web information tasks. Proceedings of the American Society for Information Science and Technology 43(1), 1–22 (2006)

[54] Keyaki, A., Miyazaki, J.: Part-of-speech tagging for web search queries using a large-scale web corpus. In: Proceedings of the Symposium on Applied Computing. pp. 931–937. ACM (2017)

[55] Kim, S.B., Han, K.S., Rim, H.C., Myaeng, S.H.: Some effective techniques for naive bayes text classification. IEEE transactions on knowledge and data engineering 18(11), 1457–1466 (2006)

[56] King, M.: Parsing natural language. Academic Press London (1983)

[57] Klein, D., Manning, C.D.: Accurate unlexicalized parsing. In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1. pp. 423–430. Association for Computational Linguistics (2003)

[58] Kolomiyets, O., Moens, M.F.: A survey on question answering technology from an information retrieval perspective. Information Sciences 181(24), 5412–5434 (2011)

[59] Koo, T., Carreras Pérez, X., Collins, M.: Simple semi-supervised dependency parsing. In: 46th Annual Meeting of the Association for Computational Linguistics. pp. 595–603 (2008)

[60] Krishnamurthy, J., Dasigi, P., Gardner, M.: Neural semantic parsing with type constraints for semi-structured tables. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. pp. 1517–1527 (2017)

[61] Labeau, M., Löser, K., Allauzen, A., von Neumann, R.J.: Non-lexical neural architecture for fine-grained pos tagging. In: EMNLP. pp. 232–237 (2015)

[62] Lai, S., Xu, L., Liu, K., Zhao, J.: Recurrent convolutional neural networks for text classification. In: AAAI. pp. 2267–2273 (2015)

[63] Lawrence, S., Giles, C.L., Fong, S.: Natural language grammatical inference with recurrent neural networks. IEEE Transactions on Knowledge and Data Engineering 12(1), 126–140 (2000)

[64] Le-Hong, P., Phan, X.H., Nguyen, T.D.: Using dependency analysis to improve question classification. In: Knowledge and Systems Engineering, pp. 653–665. Springer (2015)

[65] Lee, S.Z., Tsujii, J.i., Rim, H.C.: Part-of-speech tagging based on hidden markov model assuming joint independence. In: Proceedings of the 38th Annual Meeting on Association for Computational Linguistics. pp. 263–269. Association for Computational Linguistics (2000)

[66] Lee, U., Liu, Z., Cho, J.: Automatic identification of user goals in web search. In: Proceedings of the 14th international conference on World Wide Web. pp. 391–400. ACM (2005)

[67] Lee, Y.K., Haghighi, A., Barzilay, R.: Simple type-level unsupervised pos tagging. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing. pp. 853–861. Association for Computational Linguistics (2010)

[68] Leech, G., Svartvik, J.: A communicative grammar of English. Routledge (2013)

[69] Lewandowski, D., Drechsler, J., Mach, S.: Deriving query intents from web search engine queries. Journal of the American Society for Information Science and Technology 63(9), 1773–1788 (2012)

[70] Li, F., Zhang, X., Yuan, J., Zhu, X.: Classifying what-type questions by head noun tagging. In: Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1. pp. 481–488. Association for Computational Linguistics (2008)

[71] Li, X.: Understanding the semantic structure of noun phrase queries. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. pp. 1337–1345. Association for Computational Linguistics (2010)

[72] Li, X., Huang, X.J., WU, L.d.: Question classification using multiple classifiers. In: Proceedings of the 5th Workshop on Asian Language Resources and First Symposium on Asian Language Resources Network (2005)

[73] Li, X., Roth, D.: Learning question classifiers: the role of semantic information. Natural Language Engineering 12(03), 229–249 (2006)

[74] Li, Y., Su, L., Chen, J., Yuan, L.: Semi-supervised learning for question classification in cqa. Natural Computing 16(4), 567–577 (2017)

[75] Liu, B., Li, X., Lee, W.S., Yu, P.S.: Text classification by labeling words. In: AAAI. vol. 4, pp. 425–430 (2004)

[76] Liu, P., Qiu, X., Huang, X.: Recurrent neural network for text classification with multi-task learning. arXiv preprint arXiv:1605.05101 (2016)

[77] Liu, Y., Zhang, M., Ru, L., Ma, S.: Automatic query type identification based on click through information. In: Asia Information Retrieval Symposium. pp. 593–600. Springer (2006)

[78] Liu, Z., Jansen, B.J.: Identifying and predicting the desire to help in social question and answering. Information Processing & Management 53(2), 490–504 (2017)

[79] Liu, Z., Jansen, B.J.: Questioner or question: Predicting the response rate in social question and answering on sina weibo. Information Processing & Management 54(2), 159–174 (2018)

[80] Luneva, E., Banokin, P., Zamyatina, V., Ivantsov, S.: Natural language text parsing for social network user sentiment analysis based on fuzzy sets. In: Mechanical Engineering, Automation and Control Systems (MEACS), 2015 International Conference on. pp. 1–5. IEEE (2015)

[81] Lv, L., Liu, Y.S.: Research of english text classification methods based on semantic meaning. In: 2005 International Conference on Information and Communication Technology. pp. 689–700. IEEE (2005)

[82] Manshadi, M., Li, X.: Semantic tagging of web search queries. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2. pp. 861–869. Association for Computational Linguistics (2009)

[83] Marcus, M.P., Marcinkiewicz, M.A., Santorini, B.: Building a large annotated corpus of english: The penn treebank. Computational linguistics 19(2), 313–330 (1993)

[84] May, R., Steinberg, A.: Al, building a question classifier for a trec-style question answering system. AL: The Stanford Natural Language Processing Group, Final Projects (2004)

[85] McClosky, D., Charniak, E., Johnson, M.: Automatic domain adaptation for parsing. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. pp. 28–36. Association for Computational Linguistics (2010)

[86] Mendoza, M., Zamora, J.: Identifying the intent of a user query using support vector machines. In: International Symposium on String Processing and Information Retrieval. pp. 131–142. Springer (2009)

[87] Metzler, D., Croft, W.B.: Analysis of statistical question classification for fact-based questions. Information Retrieval 8(3), 481–504 (2005)

[88] Mishra, M., Mishra, V.K., Sharma, H.: Question classification using semantic, syntactic and lexical features. International Journal of Web & Semantic Technology 4(3), 39 (2013)

[89] Mitchell, T.M.: Machine learning. McGraw hill (1997)

[90] Mohasseb, A., Bader-El-Den, M., Cocea, M.: Analysis of the syntactical structure of web queries. In: Machine Learning and Cybernetics (ICMLC), 2018 International Conference on. IEEE (2018)

[91] Mohasseb, A., Bader-El-Den, M., Cocea, M.: Detecting question intention using a k-nearest neighbor based approach. In: IFIP International Conference on Artificial Intelligence Applications and Innovations. pp. 101–111. Springer (2018)

[92] Mohasseb, A., Bader-El-Den, M., Cocea, M.: Question categorization and classification using grammar based approach. Information Processing & Management (2018)

[93] Mohasseb, A., Bader-El-Den, M., Cocea, M., Liu, H.: Improving imbalanced question classification using structured smote based approach. In: Machine Learning and Cybernetics (ICMLC), 2018 International Conference on. IEEE (2018)

[94] Mohasseb, A., Bader-El-Den, M., Kanavos, A., Cocea, M.: Web queries classification based on the syntactical patterns of search types. In: International Conference on Speech and Computer. pp. 809–819. Springer (2017)

[95] Mohasseb, A., Bader-El-Den, M., Liu, H., Cocea, M.: Domain specific syntax based approach for text classification in machine learning context. In: Machine Learning and Cybernetics (ICMLC), 2017 International Conference on. vol. 2, pp. 658–663. IEEE (2017)

[96] Mohasseb, A., El-Sayed, M., Mahar, K.: Automated identification of web queries using search type patterns. In: WEBIST (2). pp. 295–304 (2014)

[97] Mohd, M., Hashmy, R.: Question classification using a knowledge-based semantic kernel. In: Soft Computing: Theories and Applications, pp. 599–606. Springer (2018)

[98] Moldovan, D., Paşca, M., Harabagiu, S., Surdeanu, M.: Performance issues and error analysis in an open-domain question answering system. ACM Transactions on Information Systems (TOIS) 21(2), 133–154 (2003)

[99] Morrison, J.B., Pirolli, P., Card, S.K.: A taxonomic analysis of what world wide web activities significantly impact people's decisions and actions. In: CHI'01 extended abstracts on Human factors in computing systems. pp. 163–164. ACM (2001)

[100] Nguyen, D.Q., Dras, M., Johnson, M.: A novel neural network model for joint pos tagging and graph-based dependency parsing. arXiv preprint arXiv:1705.05952 (2017)

[101] Nguyen, T.T., Nguyen, L.M., Shimazu, A.: Improving the accuracy of question classification with machine learning. In: Research, Innovation and Vision for the Future, 2007 IEEE International Conference on. pp. 234–241. IEEE (2007)

[102] Nguyen, T.T., Nguyen, L.M., Shimazu, A.: Using semi-supervised learning for question classification. vol. 3, pp. 112–130. Information and Media Technologies Editorial Board (2008)

[103] Nijholt, A.: Context-free grammars: covers, normal forms, and parsing. No. 93, Springer Science & Business Media (1980)

[104] Nithya, K., Kalaivaani, P.D., Thangarajan, R.: An enhanced data mining model for text classification. In: 2012 International Conference on Computing, Communication and Applications. pp. 1–4. IEEE (2012)

[105] Nivre, J., Hall, J., Nilsson, J.: Maltparser: A data-driven parser-generator for dependency parsing. In: Proceedings of LREC. vol. 6, pp. 2216–2219 (2006)

[106] Nivre, J., Hall, J., Nilsson, J., Chanev, A., Eryigit, G., Kübler, S., Marinov, S., Marsi, E.: Maltparser: A language-independent system for data-driven dependency parsing. Natural Language Engineering 13(2), 95–135 (2007)

[107] Pass, G., Chowdhury, A., Torgeson, C.: A picture of search. In: InfoScale - Proceedings of the 1st international conference on Scalable information systems. ACM Press, New York (2006)

[108] Peng, L., Gao, Y., Yang, Y.: Automatic text classification based on knowledge tree. In: 2008 IEEE Conference on Cybernetics and Intelligent Systems. pp. 681–684. IEEE (2008)

[109] Peters, R.A.: A linguistic history of English. Houghton Mifflin (1968)

[110] Petrov, S., Das, D., McDonald, R.: A universal part-of-speech tagset. arXiv preprint arXiv:1104.2086 (2011)

[111] Pinter, Y., Reichart, R., Szpektor, I.: Syntactic parsing of web queries with question intent. In: HLT-NAACL. pp. 670–680 (2016)

[112] Quinlan, J.R.: Induction of decision trees. Machine learning 1(1), 81–106 (1986)

[113] Quinlan, J.R.: C4. 5: programs for machine learning. Elsevier (2014)

[114] Ratnaparkhi, A.: Learning to parse natural language with maximum entropy models. Machine learning 34(1-3), 151–175 (1999)

[115] Ratnaparkhi, A., et al.: A maximum entropy model for part-of-speech tagging. In: Proceedings of the conference on empirical methods in natural language processing. vol. 1, pp. 133–142. Philadelphia, PA (1996)

[116] Rennie, J.D., Shih, L., Teevan, J., Karger, D.R., et al.: Tackling the poor assumptions of naive bayes text classifiers. In: ICML. vol. 3, pp. 616–623. Washington DC) (2003)

[117] Roa, S., Nino, F.: Classification of natural language sentences using neural networks. In: FLAIRS Conference. pp. 444–449 (2003)

[118] Roark, B., Bacchiani, M.: Supervised and unsupervised pcfg adaptation to novel domains. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1. pp. 126–133. Association for Computational Linguistics (2003)

[119] Rose, D.E., Levinson, D.: Understanding user goals in web search. In: Proceedings of the 13th international conference on World Wide Web. pp. 13–19. ACM (2004)

[120] Sagara, T., Hagiwara, M.: Natural language neural network and its application to question-answering system. Neurocomputing 142, 201–208 (2014)

[121] Saha Roy, R.: Analyzing linguistic structure of web search queries. In: Proceedings of the 22nd International Conference on World Wide Web. pp. 395–400. ACM (2013)

[122] Schmid, H.: Probabilistic part-ofispeech tagging using decision trees. In: New methods in language processing. p. 154 (2013)

[123] Schnabel, T., Schütze, H.: Flors: Fast and simple domain adaptation for part-of-speech tagging. Transactions of the Association for Computational Linguistics 2, 15–26 (2014)

[124] Sebastiani, F.: Machine learning in automated text categorization. ACM computing surveys (CSUR) 34(1), 1–47 (2002)

[125] Smith, N.A., Heilman, M., Hwa, R.: Question generation as a competitive undergraduate course project. In: Proceedings of the NSF Workshop on the Question Generation Shared Task and Evaluation Challenge (2008)

[126] Socher, R., Bauer, J., Manning, C.D., et al.: Parsing with compositional vector grammars. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). vol. 1, pp. 455–465 (2013)

[127] Socher, R., Manning, C.D., Ng, A.Y.: Learning continuous phrase representations and syntactic parsing with recursive neural networks. In: Proceedings of the NIPS-2010 Deep Learning and Unsupervised Feature Learning Workshop. pp. 1–9 (2010)

[128] Song, R., Dou, Z., Hon, H.W., Yu, Y.: Learning query ambiguity models by using search logs. Journal of Computer Science and Technology 25(4), 728–738 (2010)

[129] Song, W., Wenyin, L., Gu, N., Quan, X., Hao, T.: Automatic categorization of questions for user-interactive question answering. Information Processing & Management 47(2), 147–156 (2011)

[130] Stenetorp, P.: Transition-based dependency parsing using recursive neural networks. In: NIPS Workshop on Deep Learning. Citeseer (2013)

[131] Stratos, K., Collins, M., Hsu, D.: Unsupervised part-of-speech tagging with anchor hidden markov models. Transactions of the Association for Computational Linguistics 4, 245–257 (2016)

[132] Suganya, S., Gomathi, C., et al.: Syntax and semantics based efficient text classification framework. International Journal of Computer Applications 65(15) (2013)

[133] Sun, X., Wang, H., Xiao, Y., Wang, Z.: Syntactic parsing of web queries. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. pp. 1787–1796 (2016)

[134] Sushmita, S., Piwowarski, B., Lalmas, M.: Dynamics of Genre and Domain Intents, pp. 399–409. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)

[135] Taboada, M., Brooke, J., Tofiloski, M., Voll, K., Stede, M.: Lexicon-based methods for sentiment analysis. Computational linguistics 37(2), 267–307 (2011)

[136] Titov, I., Henderson, J.: Porting statistical parsers with data-defined kernels. In: Proceedings of the Tenth Conference on Computational Natural Language Learning. pp. 6–13. Association for Computational Linguistics (2006)

[137] Toutanova, K., Klein, D., Manning, C.D., Singer, Y.: Feature-rich part-of-speech tagging with a cyclic dependency network. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1. pp. 173–180. Association for Computational Linguistics (2003)

[138] Toutanova, K., Manning, C.D.: Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In: Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics-Volume 13. pp. 63–70. Association for Computational Linguistics (2000)

[139] Tsukuda, K., Sakai, T., Dou, Z., Tanaka, K.: Estimating Intent Types for Search Result Diversification, pp. 25–37. Springer Berlin Heidelberg, Berlin, Heidelberg (2013), http://dx.doi.org/10.1007/978-3-642-45068-6_3

[140] Tur, G., Jeong, M., Wang, Y.Y., Hakkani-Tür, D., Heck, L.: Exploiting the semantic web for unsupervised natural language semantic parsing (2012)

[141] Ture, F., Jojic, O.: Simple and effective question answering with recurrent neural networks. arXiv preprint arXiv:1606.05029 (2016)

[142] Uysal, A.K.: An improved global feature selection scheme for text classification. Expert systems with Applications 43, 82–92 (2016)

[143] Uysal, A.K., Gunal, S.: A novel probabilistic feature selection method for text classification. Knowledge-Based Systems 36, 226–235 (2012)

[144] Van-Tu, N., Anh-Cuong, L.: Improving question classification by feature extraction and selection. Indian Journal of Science and Technology 9(17) (2016)

[145] Verberne, S., van der Heijden, M., Hinne, M., Sappelli, M., Koldijk, S., Hoenkamp, E., Kraaij, W.: Reliability and validity of query intent assessments. Journal of the American Society for Information Science and Technology 64(11), 2224–2237 (2013), http://dx.doi.org/10.1002/asi.22948

[146] Wang, C., Song, Y., Li, H., Zhang, M., Han, J.: Knowsim: A document similarity measure on structured heterogeneous information networks. In: Data Mining (ICDM), 2015 IEEE International Conference on. pp. 1015–1020. IEEE (2015)

[147] Wang, C., Song, Y., Li, H., Zhang, M., Han, J.: Text classification with heterogeneous information network kernels. In: AAAI. pp. 2130–2136 (2016)

[148] Wang, P., Xu, J., Xu, B., Liu, C., Zhang, H., Wang, F., Hao, H.: Semantic clustering and convolutional neural network for short text categorization. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers). vol. 2, pp. 352–357 (2015)

[149] Wei, G., Gao, X., Wu, S.: Study of text classification methods for data sets with huge features. In: Industrial and Information Systems (IIS), 2010 2nd International Conference on. vol. 1, pp. 433–436. IEEE (2010)

[150] Wu, D., Zhang, Y., Zhao, S., Liu, T.: Identification of web query intent based on query text and web knowledge. In: Pervasive Computing Signal Processing and Applications (PCSPA), 2010 First International Conference on. pp. 128–131. IEEE (2010)

[151] Xu, S., Cheng, G., Kong, F.: Research on question classification for automatic question answering. In: Asian Language Processing (IALP), 2016 International Conference on. pp. 218–221. IEEE (2016)

[152] Yang, K., Cai, Y., Huang, D., Li, J., Zhou, Z., Lei, X.: An effective hybrid model for opinion mining and sentiment analysis. In: Big Data and Smart Computing (BigComp), 2017 IEEE International Conference on. pp. 465–466. IEEE (2017)

[153] Yang, L., Zhang, M., Liu, Y., Yu, N., Sun, M., Fu, G.: Joint pos tagging and dependency parsing with transition-based neural networks. arXiv preprint arXiv:1704.07616 (2017)

[154] Ye, Q., Wang, F., Li, B., Liu, Z.: Enhanced query classification with millions of fine-grained topics. In: International Conference on Web-Age Information Management. pp. 120–131. Springer (2016)

[155] Yen, S.J., Wu, Y.C., Yang, J.C., Lee, Y.S., Lee, C.J., Liu, J.J.: A support vector machine-based context-ranking model for question answering. Information Sciences 224, 77–87 (2013)

[156] Yih, W.t., Chang, M.W., He, X., Gao, J.: Semantic parsing via staged query graph generation: Question answering with knowledge base. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). vol. 1, pp. 1321–1331 (2015)

[157] Yu, Z., Chen, H., Liu, J., You, J., Leung, H., Han, G.: Hybrid-nearest neighbor classifier. IEEE transactions on cybernetics 46(6), 1263–1275 (2016)

[158] Zhang, B., Marin, A., Hutchinson, B., Ostendorf, M.: Learning phrase patterns for text classification. IEEE transactions on audio, speech, and language processing 21(6), 1180–1189 (2013)

[159] Zhang, D., Lee, W.S.: Question classification using support vector machines. In: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval. pp. 26–32. ACM (2003)

[160] Zhang, S., Pan, X.: A novel text classification based on mahalanobis distance. In: Computer Research and Development (ICCRD), 2011 3rd International Conference on. vol. 3, pp. 156–158. IEEE (2011)

[161] Zhang, Y., Clark, S.: A tale of two parsers: investigating and combining graph-based and transition-based dependency parsing using beam-search. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. pp. 562–571. Association for Computational Linguistics (2008)

Appendices

Appendix A: Query Grammar terms and corresponding abbreviations

Category Name	Abbreviation
Verbs	V
Action Verbs	AV
Action Verb-Interact terms	AV_I
Action Verb-Locate	AV_L
Action Verb- Download	AV_D
Auxiliary Verb	$AuxV$
Linking Verbs	LV
Adjective Free	Adj_F
Adjective Online	Adj_O
Adjective	Adj
Adverb	Adv
Determiner	D
Conjunction	$Conj$
Preposition	P
Domain Suffix	DS
Domain Prefixe	DP
Noun	N
Pronoun	$Pron$
Numeral Numbers	NN
Ordinal Numbers	NN_O
Cardinal Numbers	NN_C
Proper Nouns	PN
Celebrities Name	PN_C
Entertainment	PN_{Ent}
Newspapers, Magazines, Documents, Books	PN_{BDN}
Events	PN_E
Companies Name	PN_{CO}
Geographical Areas	PN_G
Places and Buildings	PN_{PB}
Institutions, Associations, Clubs, Parties, Foundations and Organizations	PN_{IOG}
Brand Names	PN_{BN}
Software and Applications	PN_{SA}
Products	PN_P
History and News	PN_{HN}
Religious Terms	PN_R
Holidays, Days, Months	PN_{HMD}
Health Terms	PN_{HLT}
Science Terms	PN_S
Common Noun	CN
Common Noun – Other- Singular	CN_{OS}
Common Noun- Other- Plural	CN_{OP}
Database and Servers	CN_{DBS}
Advice	CN_A
Download	CN_D
Entertainment	CN_{Ent}
File Type	CN_{File}
Informational Terms	CN_{IFT}
Obtain Offline	CN_{OF}
Obtain Online	CN_{OO}
History and News	CN_{HN}
Interact terms	CN_I
Locate	CN_L
Site, Website, URL	CN_{SWU}
Question Words	QW
How	QW_{How}
What	QW_{What}
When	QW_{When}
Where	QW_{Where}
Who	QW_{Who}
Which	QW_{Which}

Appendix B: Question Grammar terms and corresponding abbreviations

Category Name	Abbreviation
Verbs	V
Action Verbs	AV
Auxiliary Verb	$AuxV$
Linking Verbs	LV
Adjective	Adj
Adverb	Adv
Determiner	D
Conjunction	$Conj$
Preposition	P
Noun	N
Pronoun	$Pron$
Numeral Numbers	NN
Ordinal Numbers	NN_O
Cardinal Numbers	NN_C
Proper Nouns	PN
Celebrities Name	PN_C
Entertainment	PN_{Ent}
Newspapers, Magazines, Documents, Books	PN_{BDN}
Events	PN_E
Companies Name	PN_{CO}
Geographical Areas	PN_G
Places and Buildings	PN_{PB}
Institutions, Associations, Clubs, Foundations and Organizations	PN_{IOG}
Brand Names	PN_{BN}
Software and Applications	PN_{SA}
Products	PN_P
History and News	PN_{HN}
Religious Terms	PN_R
Holidays, Days, Months	PN_{HMD}
Health Terms	PN_{HLT}
Science Terms	PN_S
Common Noun	CN
Common Noun – Other- Singular	CN_{OS}
Common Noun- Other- Plural	CN_{OP}
Database and Servers	CN_{DBS}
Advice	CN_A
Entertainment	CN_{Ent}
History and News	CN_{HN}
Site, Website, URL	CN_{SWU}
Health Terms	CN_{HLT}
Question Words	QW
How	QW_{How}
What	QW_{What}
When	QW_{When}
Where	QW_{Where}
Who	QW_{Who}
Which	QW_{Which}

Appendix C: Research Ethics Review

FORM UPR16

Research Ethics Review Checklist

Please include this completed form as an appendix to your thesis (see the Postgraduate Research Student Handbook for more information

Postgraduate Research Student (PGRS) Information				**Student ID:**	799313
PGRS Name:	Alaa Mohasseb				
Department:	School of Computing	**First Supervisor:**	Mohamed Bader-El-Den		
Start Date: (or progression date for Prof Doc students)	**Feb 2016**				
Study Mode and Route:	Part-time ☐	MPhil ☐	MD ☐		
	Full-time ☒	PhD ☒	Professional Doctorate ☐		

Title of Thesis:	A Customizable Grammar-Based Framework for User-Intent Text Classification
Thesis Word Count: (excluding ancillary data)	32311

If you are unsure about any of the following, please contact the local representative on your Faculty Ethics Committee for advice. Please note that it is your responsibility to follow the University's Ethics Policy and any relevant University, academic or professional guidelines in the conduct of your study

Although the Ethics Committee may have given your study a favourable opinion, the final responsibility for the ethical conduct of this work lies with the researcher(s).

UKRIO Finished Research Checklist:
(If you would like to know more about the checklist, please see your Faculty or Departmental Ethics Committee rep or see the online version of the full checklist at: http://www.ukrio.org/what-we-do/code-of-practice-for-research/)

a)	Have all of your research and findings been reported accurately, honestly and within a reasonable time frame?	YES ☒ NO ☐
b)	Have all contributions to knowledge been acknowledged?	YES ☒ NO ☐
c)	Have you complied with all agreements relating to intellectual property, publication and authorship?	YES ☒ NO ☐
d)	Has your research data been retained in a secure and accessible form and will it remain so for the required duration?	YES ☒ NO ☐
e)	Does your research comply with all legal, ethical, and contractual requirements?	YES ☒ NO ☐

Candidate Statement:	
I have considered the ethical dimensions of the above named research project, and have successfully obtained the necessary ethical approval(s)	
Ethical review number(s) from Faculty Ethics Committee (or from NRES/SCREC):	3FB8-06A3-0E23-A29F-93E6-AD17-0995-B87B
If you have *not* submitted your work for ethical review, and/or you have answered 'No' to one or more of questions a) to e), please explain below why this is so:	
Signed *(PGRS)*:	**Date:** 29/5/2018