# A Customised Grammar Framework for Query Classification

**ALAA MOHASSEB[1], MOHAMED BADER-El-DEN[1] and MIHAELA COCEA[1]**
**[1]School of Computing, University of Portsmouth, United Kingdom**
**Emails:{alaa.mohasseb, mohamed.bader, mihaela.cocea}@port.ac.uk**

**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. A promising avenue for text classification is machine learning, which has been shown to perform well in a variety of applications including query classification and sentiment analysis. Many of the proposed approaches rely on the bag-of-words representation, which loses the information about the structure of the text. In this paper, we propose a Customised Grammar Framework for text classification, which exploits domain-related information and a new way to represent text as a series of syntactic categories forming syntactic patterns. The framework employs a formal grammar approach for transforming the text into the syntactic patterns representation. We applied the framework for the query classification problem and our results show that our approach outperforms previous ones in terms of classification performance.

*Keywords:*
Natural Language Processing (NLP), Information Retrieval, Text Classification, Query Classification, Machine Learning.

## 1. Introduction

In many classification real-world 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. For example, in signal processing, the same set of signal features may have different meanings (and thus, belong to different classes) depending on the sequence in which these features appear in the signal. Another example is text classification – in addition to words in the text, the syntax plays an important role in defining the meaning of the text.

Text classification is an important task in Natural Language Processing with many applications, such as web search (e.g. (Shi et al., 2016), Højgaard et al. (2016), (Wu et al., 2010), (Hernández et al., 2012)), question-answering (e.g. (Zhang & Lee, 2003), (Hardy & Cheah, 2013), (Li et al., 2017)), sentiment analysis (e.g. (Altrabsheh et al., 2014), Taboada et al. (2011), (Glorot et al., 2011), (Yang et al., 2017)). 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 (Wang et al., 2016).

The selection of distinctive features is essential for text classification (Uysal & Gunal, 2012; Uysal, 2016). 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 (Lai et al., 2015). Deep neural networks have been widely used in the area of text classification (Lawrence et al., 2000; Roa & Nino, 2003; Lai et al., 2015; Wang et al., 2015; Conneau et al., 2017; Liu et al., 2016). However, to use deep neural networks, typically a large amount of data is required (e.g. you must have a large number of feature vectors for deep learning to outperform other approaches) (Zhang et al., 2018a). In addition, it is computationally expensive to train deep neural networks (Iyyer et al., 2015; Zhang et al., 2018b,a).

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, e.g. (Ghose & Ipeirotis, 2011; Muhammad et al., 2015; Jung & Kwon, 2006; Tang et al., 2016). In order to achieve highly accurate classification models, the development of configurable classifiers, that could be customised to a given domain is crucial.

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 (e.g. (Yang et al., 2015; Jiang et al., 2016)), other approaches focused on user intent, i.e. the purpose of the search (Morrison et al., 2001; Baeza-Yates et al., 2006; Lewandowski et al., 2012).

Several taxonomies of user intent have been proposed (Morrison et al., 2001; Broder, 2002; Rose & Levinson, 2004; Kellar et al., 2006). Among these the most popular is Broder's taxonomy (Broder, 2002), which distinguishes between the following types of queries: (a) *Informational*, i.e. the intent is to find information, (b) *Navigational*, i.e. the intent is to reach a particular site, and (c) *Transactional*, i.e. the intent is to perform some web-mediated activity, e.g. buy products, find services.

Most information retrieval solutions that incorporate the classification of user intent use approaches based on bag-of-words (Baeza-Yates et al., 2006; Ashkan et al., 2009; Mendoza & Zamora, 2009) and dictionaries/lexicons (Beitzel et al., 2005; Jansen et al., 2008; Jansen & Booth, 2010). 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 and, by themselves, cannot distinguish between different types of queries. In other words, two queries with overlapping sets of terms may reflect two totally different intents. For example, the queries "Order Danielle Steel books" and "Danielle Steel books order" are very similar, but reflect different intentions - according to Broder's categories, the first query is *transactional*, while the second one is *informational*.

To address the limitation of word/term-based approaches that typically ignore the order and relations between terms within a piece of text, we propose a framework for classification that exploits the structure of the text, thus preserving both order and term relations. More specifically we propose the Customised Grammar Framework (CGF), which 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 a previous study (Mohasseb et al., 2018), a Customised Grammar Framework (CGF) for text classification was first introduced and applied for questions categorization and classification. In this study, the framework is applied to query classification according to user intent by using Broder's categories of intent (Broder, 2002). The aim is to assess the influence of using the structure of a query and the domain-specific syntactic categories on the classification performance. To achieve this aim, the following objectives are defined:

1. Investigate the influence of the different levels of detail of domain-specific information (reflected in the domain-specific syntactic categories) on the classification performance;
2. Compare the performance of different machine learning algorithms for the classification of user intent;
3. Investigate the classification performance in comparison with state-of-the art approaches.

The rest of the paper is organised as follows. Section 2 outlines previous work in query intent categorisation by outlining different query taxonomies. Section 3, as well as previous classification approaches of user intent using machine learning techniques. Section 4 describes the proposed framework, which is applied for the query classification problem in Section 5. The experiments setup and results are presented in Section 6. Section 7 provides performance comparison between our approach and other approaches, while a comparison between our approach and previous ones is discussed in Section 8. Finally, Section 9 concludes the paper and outlines directions for future work.

## 2. Categories of Queries

Different categories of web queries according to user intent were defined, which are summarised in Table 1, and discussed below.

Table 1: Summary of user intent categories for web queries

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

Web queries were classified by Morrison et al. (2001) 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 (Broder, 2002) are most commonly used in query classification. According to Broder (2002) 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 Rose & Levinson (2004) and Jansen et al. (2008) by adding sub-categories. In Rose & Levinson (2004) sub-categories were added for the informational and transactional categories, while Jansen et al. (2008) added subcategories for all three types of queries. In Lewandowski et al. (2012), Broder's categories (Broder, 2002) were extended with two others, commercial and local.

Rose & Levinson (2004) replaced the transactional queries with a category called *resource queries*, which they argue is broader than the transactional queries. The expansion of the taxonomy by Jansen et al. (2008), however, reverted the name to transactional, while keeping the subcategories initially proposed by Rose & Levinson (2004) under the name of resource queries.

In Baeza-Yates et al. (2006), 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 Kellar et al. (2006) 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 Ashkan et al. (2009), 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 (Broder, 2002), i.e. navigational and informational.

In Calderón-Benavides et al. (2010) 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 (Broder, 2002) and the one by Sushmita et al. (2010) (see below) another multi-dimensional classification was proposed by Verberne et al. (2013).

A classification according to the types of documents sought by a user was proposed in Sushmita et al. (2010), by using the domain (image/video/map) and genre (news/blogs/wikipedia). With a focus on results diversification, Bhatia et al. (2012) 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 subsection we focus mainly on query classification using Broder's categories (Broder, 2002) or their variations (Rose & Levinson, 2004; Jansen et al., 2008), as this is the most popular user intent taxonomy and our proposed framework is validated using these intent categories.

## 3. Related Studies

In the following sections we review previous work related to text classification and query classification. The different types of methods and techniques used for text classification are outlined in Section 3.1, while Section 3.2 reviews previous work on query classification methods based on Broder's categories (Broder, 2002) and using machine learning approaches.

### *3.1. Text Classification*

Many different machine learning approaches have been used to classify natural language sentences and words; Recurrent Neural Networks (RNN) is one of the approaches that have been used by many researches. In Lawrence et al. (2000) and Roa & Nino (2003), recurrent neural networks were used to classify natural language sentences as grammatical or ungrammatical. In Roa & Nino (2003), 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 (Chomsky, 1993). Lawrence et al. (2000) also examined the use of various recurrent neural network architectures like FGS, N&P, Elman, and W&Z to train a network for classification.

Lai et al. (2015) introduced a recurrent convolutional neural network for text classification without human-designed features by applying a recurrent structure to capture contextual information when learning word representations. Conneau et al. (2017) 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. In Liu et al. (2016) 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. In addition, researches used machine learning algorithms such as K-Nearest Neighbour as a mean of classification, in addition to feature selection. Basu & Murthy (2012) 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.

Nithya et al. (2012) proposed a mining model consisting of sentence, document and corpus-based concept-analysis. K-Nearest Neighbour was used for the classification process. In Liu et al. (2004), a method was proposed that combined clustering and feature selection to labels set of representative words for each class, followed by the use of these words to extract a set of documents for each class. Furthermore, Yu et al. (2016) designed the RS-HBKNN classifier in order to improve the performance of hybrid KNN (HBKNN). In Wei et al. (2010), the authors implemented a text classification system based on mutual information and K-nearest neighbour algorithm and support vector machine.

According to Zhang & Pan (2011), KNN is sensitive to the distance or similarity metric used; the typical Eucledean distance function used in classifying a test instance can cause low classification accuracy and limit the KNN classifiers utilization in text classification. A Mahalanobis distance for text classification was used and the MDKNN algorithm was developed based on the use of this distance fucntion.

Naive Bayes has also been used to automatically classify text, but according to Kim et al. (2006) 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. They proposed two empirical heuristics, i.e. per-document text normalization and a feature weighting method. Lv & Liu (2005) proposed a method based on WordNet thesaurus and Latent Semantic Indexing (LSI) model, as well as use of Naive Bayes for text classification, and a simple vector distance text classification. According to them incorporating linguistic knowledge into the text representation can lead to improvements in classification accuracy. Han et al. (2009) 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. Moreover, Gong & Yu (2010) designed and tested a system for Chinese text categorization based on the Bayes theory.

Other works for text classification, using less known approaches, are outlined in the following. Peng et al. (2008) introduced a new method for automatic text classification based on knowledge tree to simulate the process of human classification.

In Suganya et al. (2013) 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 in parallel on the syntactic and semantic levels. The outputs of these two classifiers were then combined and given as input to the third classifier. Zhang et al. (2013) 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. Finally, Wang et al. (2016) proposed a ‘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.

### 3.2. *Query Classification*

Most of the previous approaches use all three categories of Broader’s taxonomy which are summarised in Table 2, although some of them combine two of the categories (denoted by * in Table 2) into one: (a) the informational and transactional queries are grouped into one category/class in Liu et al. (2006); (b) the navigational and transactional categories are grouped together in Baeza-Yates et al. (2006) and González-Caro & Baeza-Yates (2011). Three of the previous works, i.e. Lee et al. (2005); Ashkan et al. (2009); Tsukuda et al. (2013), use the informational and navigational categories, while excluding the transactional one.

A variety of features have been used, of which the most popular are:

- past user click behaviour or click-through data (Lee et al., 2005; Liu et al., 2006; Mendoza & Zamora, 2009); a practical issue with the user-click behaviour is the accumulation of enough user clicks for a given query (Lee et al., 2005), as well as what constitutes the same query (e.g. the exact same query or a minimum overlap in the terms of the query);

- anchor text data (Lee et al., 2005; Herrera et al., 2010); research by (Liu et al., 2006) indicated that anchor text data is applicable for less than 20% of the queries, concluding that it may be applicable to some queries, but not for the majority.

- log features, e.g. IP address, user ID, time stamp, query terms (Kathuria et al., 2010; Herrera et al., 2010);

Table 2: Research using Broders categories and machine learning.

| Authors | Inf. | Nav. | Trans. |
|---|---|---|---|
| Lee et al., 2005 Lee et al. (2005) | X | X | |
| Liu et al., 2006 Liu et al. (2006) | X* | X | X* |
| Baeza-Yates et al., 2006 Baeza-Yates et al. (2006) | X | X* | X* |
| Jansen et al., 2008 Jansen et al. (2008) | X | X | X |
| Ashkan et al., 2009Ashkan et al. (2009) | X | X | |
| Mendoza et al., 2009 Mendoza & Zamora (2009) | X | X | X |
| Kathuria et al., 2010 Kathuria et al. (2010) | X | X | X |
| Herrera at al, 2010 Herrera et al. (2010) | X | X | X |
| González-Caro and Baeza-Yates, 2011 González-Caro & Baeza-Yates (2011) | X | X* | X* |
| Hernandez et al., 2012 Hernández et al. (2012) | X | X | X |
| Tsukuda et al, 2013 Tsukuda et al. (2013) | X | X | |
| Figueroa, 2015 Figueroa (2015) | X | X | X |

- user session related information, e.g. the number of times a query was reformulated per session (Kathuria et al., 2010; Mendoza & Zamora, 2009); the automatic identification of user sessions has been proven difficult (Gayo-Avello, 2009), while also presenting the issue that within the same session the user may have several intents/goals (Figueroa, 2015).

- "bag-of-words", i.e. the terms (words) are the features and the values are metrics of frequency (Baeza-Yates et al., 2006; Mendoza & Zamora, 2009; Ashkan et al., 2009; Herrera et al., 2010; González-Caro & Baeza-Yates, 2011; Figueroa, 2015); the use of bag-of-words features is very popular not just in query classification, but more broadly in text classification as well;

- PoS tags, i.e. corresponding part-of-speech (PoS), for each word or term (Hernández et al., 2012; Figueroa, 2015), typically obtained by an automatic tagger such as the Stanford one[1]; these features are also popularly used for text classification.

Other less used features are: (a) click on advertisements displayed in the results page (Ashkan et al., 2009); (b) reading time of a search result (Mendoza & Zamora, 2009); (c) linguistic-based features such as named entity, dependency trees (for representing lexical dependency) and expansion terms from WordNet (Figueroa, 2015).

The use of the features mentioned above, and in particular the bag-of-words and PoS tags, leads to large sparse datasets, which are typically reduced by removing features with low frequency.

The classification accuracy of these previous works ranges from 64.4% (Tsukuda et al., 2013) to above 90% (Lee et al., 2005; Kathuria et al., 2010). The previous work also indicated different degrees of difficulty in identifying the three types of user intent, i.e. informational, navigational and transactional. The informational queries are the most frequent and the easiest to identify, while the other two categories are less frequent and harder to identify (Hernández et al., 2012; Figueroa, 2015).

Unlike the previous approaches, we propose a formal grammar-based framework for classification, which exploits the structure within the text through a new representation using general and domain-specific syntactic categories. Details of the framework are given in the next section and its use on query classification is detailed in Section 5.

## 4. Customised Grammar Framework

We propose the Customised Grammar Framework (CGF) 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 text. The general framework is described below, while its use for the query classification problem is detailed in the following section.

[1] https://nlp.stanford.edu/software/tagger.shtml

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 CFG 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 representation 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.

Our 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*.

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

In **Phase I**, a formal grammar (see Definition 1) is defined based on the analysis of the text in conjunction with the domain knowledge for a particular problem. Domain knowledge is captured from the analysis of the given text or sentence, then this knowledge is generated automatically using the term categories and syntactic patterns resulting in the creation of the domain customized grammar. Thus, the framework can be applied to other domains automatically when a taxonomy for the domain is provided. In other words, the process of transforming the text into syntactical patterns is automatic, while the domain-specific information is captured through the taxonomy of syntactic categories.

**Definition 1.** *A grammar is a tuple* $(N, \Sigma, P, S)$*, where:*

1. *$N$ is a finite set of non-terminal symbols, which in our context are words or groups of words (e.g. 'books', 'Jane Austin');*
2. *$\Sigma$ is a finite set of terminal symbols that is disjoint from $N$ (i.e $\Sigma$ 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.*

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.

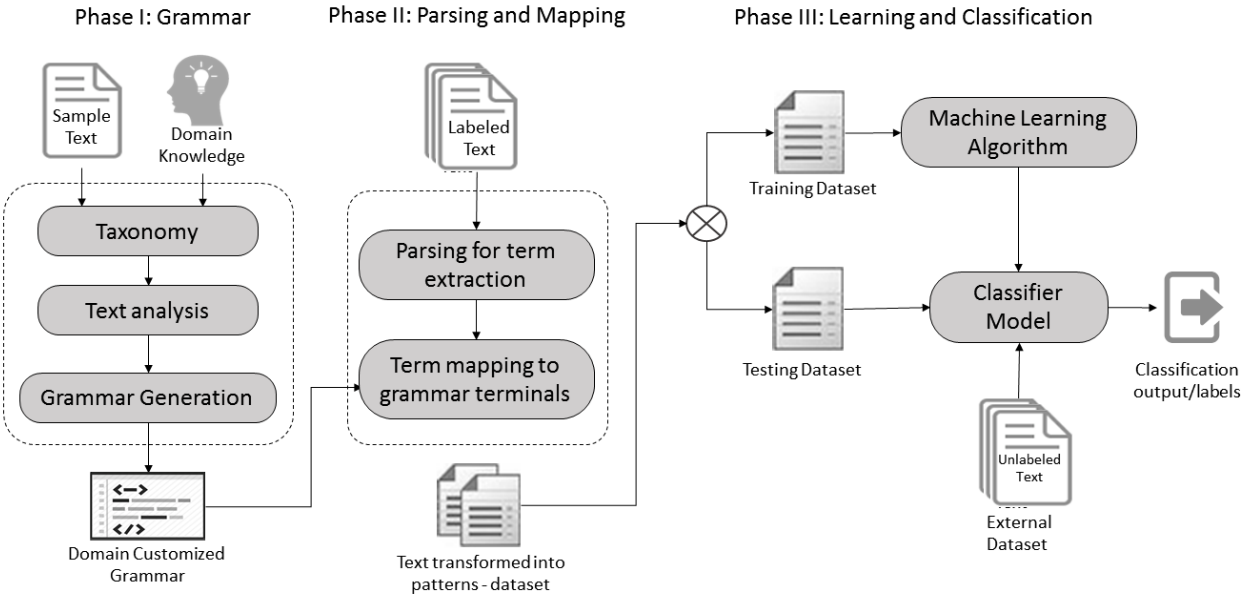
Phase I: Grammar
Phase II: Parsing and Mapping
Phase III: Learning and Classification
Sample Text
Domain Knowledge
Taxonomy
Text analysis
Grammar Generation
Domain Customized Grammar
Labeled Text
Parsing for term extraction
Term mapping to grammar terminals
Text transformed into patterns - dataset
Training Dataset
Testing Dataset
Machine Learning Algorithm
Classifier Model
Classification output/labels
Unlabeled Text
External Dataset

Figure 1: Customised Grammar Framework for text classification

The grammar is used in **Phase II** to transform the text into syntactic patterns by first tokenizing the text into a series on 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 instance “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 III** 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 the next section for the problem of query classification using Broder’s taxonomy (Broder, 2002).

## 5. CGF for Query Classification

In this section we explain in detail the use of the CGF framework and how this framework could be applied to different domains. The following subsections present each of the three phases of the framework and how they have been used in query classification.

### *5.1. Phase I: Grammar*

The CGF concept 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. In this paper, a context free grammar in the Backus normal form (BNF) is used. It has been argued (King, 1983; Peters, 1968; Nijholt, 1980) that BNF can not provide a full description of the English grammar, however, the target in this paper is to use a simple version of the English grammar combined with domain-specific syntactic categories to guide the text classification stage.

To identify the relevant syntactic categories (both general and domain-specific), the different types of queries based on Broder’s taxonomy (Broder, 2002) and Broder’s extended taxonomy (Jansen et al., 2008) were analysed, as detailed below in Sections 5.1.1 and 5.1.2, respectively. Based on the identified syntactic categories, the formal grammar is defined in Section 5.1.3.

The analysis starts with the known syntactic phrases structures and categories of the English grammar. There are seven types of phrases, of which the most used five are: Noun Phrase (*NP*), Verb Phrase (*VP*), Prepositional Phrase (*PP*), Adjectival Phrase (*AP*) and Adverbial Phrase (*AdvP*).

The syntactic categories of the English grammar are typically referred to as word-classes or part-of-speech (PoS) tags. There are 7 major word classes: Verb (*V*), Noun (*N*), Determiner (*D*), Adjective ($Adj$), Adverb (*Adv*), Preposition (*P*) and Conjunction ($Conj$). Some word classes have subclasses; for Noun, the subclasses are: Common Noun (*CN*), Proper Noun (*PN*), Pronoun (*Pron*) and Numeral Noun (*NN*); for verb the subclasses are: Action Verb ($AV$), Linking Verb (*LV*) an Auxiliary Verb (*AuxV*).

The different types of queries were analysed to be able to build the grammatical rules (e.g. types of phrases and syntactic categories), in which the domain-specific categories have been created after studying the characteristics of each query type according to work done by (Broder, 2002) and (Jansen et al., 2008). According to the authors, each of these query types has its own characteristics that would help in the identification and classification process. Using these as a starting point, the text was manually analysed to define domain-specific syntactic categories that capture these characteristics. These characteristics and their syntactic categories are described in detail in the following two subsections.

*5.1.1. Analysis of Broder's Query Types*

In this section the analysis of the syntactic characteristics of queries is described for Broder's taxonomy (Broder, 2002).

**1) Informational Query**: these consist of Phrases such as Noun phrase (*NP*), Verb phrase (*VP*), and Prepositional phrase (*PP*), e.g. *"location of Hyde Park in London"*. The most used word classes in this type are: a) Nouns, such as Common Nouns, e.g. *"county", "company", "place"* and Proper Nouns, e.g. *"England"*, *"Eiffel Tower"* and *"Adele"*; b) Question words, e.g. *"Why exercise is important?*. As this type of query is the only one to contain question words, these are important for distinguishing them from other types; thus, the syntactic category Question word (*QW*) is identified as a domain-related terminal for the formal grammar.

**2) Navigational Query**: this type of queries have a fixed grammatical structure which is the Noun Phrase (*NP*), however, the query could also simply be a web link. The only word class in this type of query is Proper Nouns (*PN*) since queries typically contain just one word, such as the name of an organisation, business, company or university, e.g."IBM". When the query takes the form of a link, the structure consists of domain suffixes and prefixes such as in *"https://www.yahoo.co.uk"* or *"ebay.com"*; consequently, the syntactic categories Domain Prefix (*DP*) and Domain Suffix ($DS$) are identifies as domain-related syntactic categories.

**3) Transactional Query**: the grammatical structure of these queries consists mostly of Verb Phrases (*VP*) and Adverbial Phrases (*AdvP*), e.g. "buy cheap phones"; also, Noun Phrases (*NP*) could be present, e.g. *"Sam Smith lyrics*. Most transactional queries include: (a) Action Verbs ($AV$), such as *"order, buy, purchase, download"*; (b) Adjectives ($Adj$) such as *"free and online"*. Typically Question words (*QW*), Pronouns (*Pron*), and Auxiliary verbs (*AuxV*) are not used in transactional queries.

*5.1.2. Analysis of the Broder's extended Query Types*

In this section the analysis of the syntactic characteristics of queries is described for Broder's extended taxonomy (Jansen et al., 2008).

***1) Informational List***: plural query terms (corresponding to the syntactic category Common Nouns Plural ($CN_P$)) 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 Atlanta"*. 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 Disney 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 above, informational list queries also include general syntactic terms such as Action verbs ($AV$), Adjectives ($Adj$), Prepositions (*Prep*), Numeral Nouns (*NN*) and Determiners (*D*).

***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. *"decoration 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 *"uTorrent"*, *"Photoshop"* and *"Skype"*, Products ($PN_P$), such as *"iPad"* and *"Oreo cookies"*, Brand Names ($PN_{BN}$), such as *"Coach"*, *"Pepsi"* and *"Gucci"*. Furthermore, word classes such as Action verbs ($AV$) and numeral nouns ($NN$) could be found in some queries.

***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 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) 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. *"Simone Biles"*, *"Vietnam war"* and *"hypertension"*. Word classes such as common noun ($CN$) and numeral noun ($NN$) are frequently used in this query type. Moreover, this the only informational category that does not have some word classes such as Question words, Pronouns, Auxiliary verbs and linking verbs.

***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 contains 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 Brazil"*, Holidays, Days and Months ($PN_{HMD}$), such as *"Christmas"*, *"Saturday"* and *"November"*. Furthermore, all question words such as when, how, where, what, who could be found in this search, e.g. *"what is a prime number?"*

***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 metals shiny"*, 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. *"honeybee communication"*.

***7) Navigational Query***: these queries typically contains 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 ($DP$).

***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 ($Adj$).

***9) Transactional Download free***: the queries in this type of search mainly consist 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 games"* and *"free mp3 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 ($PN_C$).

***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. *"safe haven book download"* and *"Kelly Clarkson 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 ($PN_P$).

***11) Transactional obtain online***: this type of queries mainly consist of common noun terms related to obtained online ($CN_{OO}$), e.g. *"meatloaf recipes"*, Entertainment ($CN_{Ent}$), such as *"Adele 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.

***12) Transactional obtain offline***: this type of queries mainly consists of common noun terms related to obtain offline ($CN_{OF}$), e.g. *"Bon Jovi wallpapers"* and *"windows 7 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.

Table 3: 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 | $PN_{SA}$, $PN_{BN}$, $PN_E$, $PN_{Ent}$, $PN_{BDN}$,$PN_G$, $PN_{IOG}$, $PN_{PB}$, $PN_{CO}$, $PN_C$, $PN_P$ |

### 5.1.3. *Customised Grammar*

In Section 4, Definition 1, we defined the formal grammar as a tuple $(N, \Sigma, P, S)$. In this section we present the details of the formal grammar 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 $\Sigma$ of terminals consists of all the syntactic categories, both general and domain-specific. We organised these in the hierarchical structure displayed in Table 3, reflecting five different levels of detail related to the syntactic categories; a list of all the syntactic categories and corresponding acronyms is displayed in the Appendix.

Below we illustrate a number of rules 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).

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

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

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

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

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

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

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

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

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

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

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

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

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

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

### *5.2. Phase II: Parsing and Mapping*

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

**Algorithm 1** Parsing and Mapping Algorithm

```
Read query $q$ 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
```

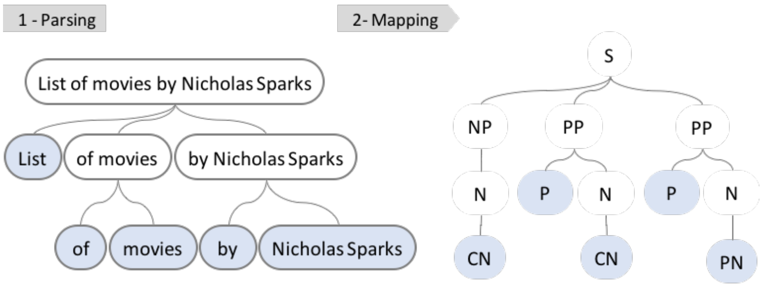

Figure 2: Phase II: Parsing and Mapping example

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 our experiments this maximum length was 13. For patterns of lower length, some attributes will have no values; for example, the pattern in the example above 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.

### *5.3. Phase III: Learning and Classification*

In this phase the patterns 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 on unlabelled queries.

We used several learning algorithms and evaluated their performance, as outlined in the Experiments section below.

## 6. Experiments

In this section we present two sets of experiments conducted to achieve the objectives outlined in Section 1. For the first objective, i.e. investigate the influence of the different grammar terminals levels of detail on the classification accuracy, we ran experiments with different versions of the grammar, corresponding to the five levels for the terminals set; these experiments are described in subsection 6.1. To validate the findings from the experiments related to the levels of detail for the grammar, we ran another set of experiments, which are outlined in subsection 6.2.

For both sets of experiments, four machine learning algorithms were used: (1) decision trees, and in particular the J48 implementation in Weka; (2) Random Forest, (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-score. We investigated the classification of queries according to Broder's categories (i.e. 3-class models), as well as Broder'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 8, where we discuss the results of our approach in comparison with previous ones.

Table 4: 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 |

### *6.1. Experiments on grammar levels*

For this experiment, the 1953 labelled queries from Mendoza & Zamora (2009) were used, and 4047 queries were randomly selected from the AOL 2006 dataset (Pass et al., 2006) and labelled according to the procedure described in Mohasseb et al. (2014). From the 4047 AOL queries, 116 were vague or contained mistakes and thus, were excluded, leading to 5884 queries used in the experiments. Their distribution according to Broder's taxonomy and Broder's extended taxonomy is given in Table 4.

The evaluation metrics for the 3-class models resulting from the four learning algorithms for each level of the grammar are displayed in Table 5. In addition to the overall performance, precision, recall and F-score 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 significant 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 6. These show similar results as for the 3-class models, with significant improvement 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.

Table 5: Performance of the classifiers for Informational (Info.), Navigational (Nav.) and Transactional (Trans.) queries (3-class models)

| | | $CGF_{JRip}$ | | | $CGF_{RF}$ | | | $CGF_{J48}$ | | | $CGF_{NB}$ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| L1 | Accuracy | 55.11% | | | **66.26%** | | | 66.02% | | | 58.85% | | |
| | Precision | 0.53 | | | 0.85 | | | 0.84 | | | 0.87 | | |
| | Recall | 0.94 | | | 0.69 | | | 0.69 | | | 0.53 | | |
| | F-score | 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-score | 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-score | 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 | 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-score | 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 | 0.99 | 1.00 | 1.00 | 0.99 | 0.99 | 0.93 | 0.84 | 0.88 |
| | Nav. | 0.96 | 1 | 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-score: | 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 |

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 ($CGF_{JRip}$, $CGF_{RF}$ and $CGF_{J48}$) are unable to identify navigational queries, and only the Naive Bayes classifier is able to correctly identity some of the 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

Table 6: Performance of the 12-class models.

| | $CGF_{JRip}$ | | | | $CGF_{RF}$ | | | | $CGF_{J48}$ | | | | $CGF_{NB}$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 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 |

between the navigational queries and the other two types, i.e. informational and transactional.

When subcategories of the English main syntactic categories are used, i.e. level 2, we see a dramatic improvement 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 identified 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 ($CGF_{RF}$, $CGF_{J48}$ and $CGF_{NB}$), while for $CGF_{JRip}$ 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.

Table 7: 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 | $AV_I$, $AV_L$, $AV_D$, $NN_C$, $NN_O$, $QW_Who$, $QW_{What}$, $QW_{Where}$, $QW_{When}$, $QW_{How}$, $QW_{Which}$, $DS$, $DP$, $PN_C$, $PN_S$, $PN_{HLT}$, $PN_{HMD}$, $PN_R$, $PN_{HN}$, $PN_{SA}$,$PN_{BN}$, $PN_E$, $PN_{Ent}$, $PN_{BDN}$, $PN_G$, $PN_{IOG}$, $PN_{PB}$, $PN_{CO}$, $CN_A$, $CN_{SWU}$, $CN_D$, $CN_{HN}$, $CN_{OS}$, $CN_{OP}$ $CN_I$,$CN_L$, $CN_{OB}$, $CN_{EFI}$. |

The performance of all classifiers for all classes improves further at level 4, which has more details related to the types of queries from Broder's extended categories. There is an improvement even for the navigational queries, although there are no sub-types for the navigational queries in Broder'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 6), 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 Broder'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.

Table 8: 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 |

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 subcategories (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, we conducted a new set of experiments, which is described in the next subsection.

### *6.2. 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 7. We denote the new levels as L1, L2 and L3 to distinguish them from the previous grammar structure denoted by levels 1 to 5.

This modification resulted in the exclusion of 10 syntactic categories from levels 3 and 4 that contain subcategories 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 subcategories will become subcategories 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 subcategories 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 Mohasseb et al. (2014, 2017a,b). These were used together with the 1953 labelled queries from Mendoza & Zamora (2009) – thus, 10,000 queries were used, which are distributed as outlined in Table 8.

The results for the 3-class models are given in Table 9 and for the 12-class models in Table 10; the results per class using level L3 and Random Forest for the 12-class models are given in Table 11. 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 5), with slight variations which are likely due to the variation in the data used.

For level L3, the performance is similar to the results for level 5 in the previous structure (see Table 5), as both of these levels contain all the domain-specific syntactic categories.

In the following, we discuss the results in relation to the objectives outlined in Section 1.

Our **first objective** was to investigate the optimal level of detail for the domain-related syntactic categories. The results from the experiments in Sections 6.1 and 6.2 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

Table 9: Performance of the classifiers for Informational, Navigational and Transactional queries (3-class models).

| | | $CGF_{JRip}$ | | | $CGF_{RF}$ | | | $CGF_{J48}$ | | | $CGF_{NB}$ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| L1 | Accuracy | 59.5% | | | **63.4%** | | | 63.3% | | | 53.71% | | |
| | Precision | 0.53 | | | 0.61 | | | 0.61 | | | 0.67 | | |
| | Recall | 0.60 | | | 0.63 | | | 0.633 | | | 0.537 | | |
| | F-score | 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-score | 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-score | 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 10: Performance of the 12-class models.

| | $CGF_{JRip}$ | | | | $CGF_{RF}$ | | | | $CGF_{J48}$ | | | | $CGF_{NB}$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *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 |

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.

The **second objective** was about which machine learning algorithms are best suited to classification of user intent, when using the data representation proposed in the CGF framework. $CGF_{NB}$, which is known to perform well on textual data, leads to the lowest performance models in our experiments (but not by much), while $CGF_{RF}$ leads to the best performing model. When using the domain-specific syntactic categories (levels 3, 4 and 5 in Tables 5 and 6, and level L3 in Tables 9 and 10) JRip and J48 are very close in performance to $CGF_{RF}$, especially at level 5 in Table 5 and level L3 in Table 9. 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 **third objective** was about the classification performance of our approach in comparison with state-of-the-art approaches. This is discussed in detail in the following section.

## 7. Performance comparison

In this section experiments have been conducted for the objective of validating our proposed approach in improving the classification accuracy and the identification of different type of queries and to compare the classification performance of our approach with the state-of-the-art approaches.

### *7.1. CGF vs. n-gram*

In this section experiments have been conducted using the typical bag-of-words representation, through the use of n-grams. The data was pre-processes by removing stop words and using the Snowball Stemmer. The classifiers were

Table 11: 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 |

been built using the Knime software [2].

From the previous experiments, the selected the best two machine learning algorithms, i.e. J48 and RandomForests (RF). Similar to previous experiments, to assess the performance of the machine learning classifiers the experiments were set up using the typical 10-fold cross-validation.

Table 12 presents the classification performance results (Precision, Recall and F-Measure) of the n-gram$_J$48 and n-gram$_{RF}$ classifiers when using Broder's query categories, i.e. the three-class dataset. The results show that, when using the n-grams as features, the decision tree (n-gram$_{J48}$) identified correctly (i.e. Recall) 90.9% of the queries, while the random forest (n-gram$_{RF}$) had a recall of 95.2%. In addition, Table 13 presents the classification performance results (Precision, Recall and F-Measure) of the n-gram$_{J48}$ and n-gram$_{RF}$ using Broder's extended query categories, i.e. the 12-class dataset. The results show that the decision tree (n-gram$_{J48}$) identified correctly (i.e. Recall) 94.1% of the queries, while the random forest (n-gram$_{RF}$) correctly identifies 92.4% of the queries.

Table 12: Performance of the classifiers using Broder's categories and the features and n-gram framework - $CGF_{RF}$ results are highlighted in bold. Precision (P), Recall (R), F-Measure (F).

| | $CGF_{RF}$ | | | n-gram$_{RF}$ | | | n-gram$_{J48}$ | | |
|---|---|---|---|---|---|---|---|---|---|
| Accuracy: | **99.9%** | | | 95.2 % | | | 90.9% | | |
| Class: | 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 |
| Nav. | **1.00** | **1.00** | **1.00** | 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 |

These results validate that using domain-specific information and preserving the structure of the query improve the classification accuracy and could be used for the identification of informational, navigational and transactional queries, in addition to the extended categories of these queries. Furthermore, even though using n-grams as features with the typical text preprocessing could be used for the classification of informational, navigational and transactional queries, it could not be used for the classification of most extended categories. Informational queries extended categories such as undirected, directed-open and directed-closed had 0 precision, recall and F-Measure for both classifier. Similarly, the extended categories of the navigational type of queries had 0 precision, recall and F-Measure for both classifiers. Furthermore, some transactional queries from the extended categories had low precision and recall, e.g. transactional download free and transactional obtain-offline.

[2] http://knime.com

Table 13: Performance of the classifiers using Broder's extended categories - $CGF_{RF}$ results are highlighted in bold. Precision (P), Recall (R), F-Measure (F).

| | $CGF_{RF}$ | | | n-gram$_{RF}$ | | | n-gram$_{J48}$ | | |
|---|---|---|---|---|---|---|---|---|---|
| Accuracy: | **99.6%** | | | 92.4% | | | 94.1% | | |
| Class: | 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 |
| Info. Advice | **0.99** | **0.99** | **0.99** | 1.00 | 0.97 | 0.99 | 1.00 | 0.97 | 0.98 |
| Info. List | **0.99** | **1.00** | **0.99** | 1.00 | 0.95 | 0.97 | 1.00 | 0.95 | 0.97 |
| Info. Directed Open | **0.98** | **0.92** | **0.95** | 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 |
| Info. Find | **0.99** | **0.99** | **0.99** | 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 |
| Tran. Download Free | **1.00** | **0.98** | **0.99** | 0.62 | 0.95 | 0.75 | 0.99 | 0.83 | 0.90 |
| Tran. Download not Free | **1.00** | **0.99** | **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 |
| Tran. Obtain offline | **0.99** | **1.00** | **0.99** | 0.97 | 0.47 | 0.63 | 0.64 | 1.00 | 0.78 |
| Tran. Obtain Online | **1.00** | **0.99** | **1.00** | 0.99 | 0.98 | 0.99 | 0.99 | 0.98 | 0.99 |

### *7.2. CGF vs. Neural Networks*

In this section experiments have been conducted using Neural Networks (NN), to compare the their performance with out proposed approach. Similar to the previous experiments to assess the performance of the machine learning classifier the experiments were set up using the typical 10-fold cross validation. The Deep Neural Networks (DNN) implementation in Weka was used; the network consists of three layers and the word embedding 'word2vec' approach was used to convert each word in the query to a vector. Word2vec takes as its input the query (input layer) to produce a vector space which means that each unique word in the query will be assigned a corresponding vector in the vector space. A Long short-term memory (LSTM) (Hochreiter & Schmidhuber, 1997) type of network was used with one hidden layer. In addition, the stochastic gradient descent algorithm was used for learning optimization.

Table 14 presents the classification performance results (precision, recall and F-Measure) of neural networks classifier using Broder's query categories. The results show that neural network identified correctly (i.e. recall) 96.1% of the queries. In addition, Table 15 presents the classification performance results of the NN classifier using Broder's extended query categories. The results show that the NN identified correctly (i.e. recall) 90.9% of the queries.

Table 14: Performance of the classifiers using Broder's categories and Neural Networks - $CGF_{RF}$ results are highlighted in bold. Precision (P), Recall (R), F-Measure (F).

| | $CGF_{RF}$ | | | NN | | |
|---|---|---|---|---|---|---|
| Accuracy: | **99.9%** | | | 96.1 % | | |
| Class: | P | R | F | P | R | F |
| Info. | **1.00** | **1.00** | **1.00** | 0.98 | 0.96 | 0.97 |
| Nav. | **1.00** | **1.00** | **1.00** | 0.96 | 0.99 | 0.98 |
| Tran. | **1.00** | **1.00** | **1.00** | 0.54 | 0.19 | 0.28 |

The 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. Furthermore, even though the neural network algorithm had a good overall performance when applying it to Broder's query taxonomy, it has achieved a low precision, recall and F-measure for transactional queries; the instances from this category have been mostly miss-classified as informational and navigational. In addition, for NN, the informational queries extended categories such as advice, directed-open and directed-closed had lower recall compared with the other categories. Consequently, $CGF_{RF}$ has better overall results than NN with Broder's extended taxonomy.

Table 15: Performance of the classifiers using broder's extended categories - $CGF_{RF}$ results are highlighted in bold. Precision (P), Recall (R), F-Measure (F).

| | $CGF_{RF}$ | | | NN | | |
|---|---|---|---|---|---|---|
| Accuracy: | **99.6%** | | | 90.9% | | |
| Class: | P | R | F | P | R | F |
| Info. undirected | **1.00** | **1.00** | **1.00** | 1.00 | 0.90 | 0.95 |
| Info. Advice | **0.99** | **0.99** | **0.99** | 0.93 | 0.79 | 0.85 |
| Info. List | **0.99** | **1.00** | **0.99** | 0.90 | 0.97 | 0.93 |
| Info. Directed Open | **0.98** | **0.92** | **0.95** | 0.96 | 0.61 | 0.74 |
| Info. Directed Closed | **0.98** | **0.99** | **0.99** | 0.93 | 0.69 | 0.79 |
| Info. Find | **0.99** | **0.99** | **0.99** | 0.99 | 0.89 | 0.94 |
| Nav. | **0.99** | **1.00** | **1.00** | 0.97 | 0.92 | 0.94 |
| Tran. Download Free | **1.00** | **0.98** | **0.99** | 0.98 | 0.86 | 0.92 |
| Tran. Download not Free | **1.00** | **0.99** | **0.99** | 0.87 | 0.90 | 0.89 |
| Tran. Interact | **0.99** | **1.00** | **0.99** | 0.76 | 0.92 | 0.83 |
| Tran. Obtain offline | **0.99** | **1.00** | **0.99** | 0.98 | 0.90 | 0.94 |
| Tran. Obtain Online | **1.00** | **0.99** | **1.00** | 0.99 | 0.98 | 0.98 |

## 8. Discussion

In this section we discuss the performance of previous research; we summarise the performance on previous automatic classification approaches in Table 16 (where several models are reported, e.g. with feature variations, we report the best performance). With the exception of Jansen et al. (2008), which adopted a rule-based approach, all other approaches use machine learning. For Baeza-Yates et al. (2006), the values in the table are approximate numbers, as in the original paper they were displayed in a graph.

In terms of accuracy, the highest performance is obtained by Lee et al. (2005), i.e. 90%, and Kathuria et al. (2010), i.e. 94%. A classification approach was used by Lee et al. (2005) through linear regression, while Kathuria et al. (2010) used a clustering approach through the k-means algorithm. Neither of these two works report performance by class. Our approach leads to over 99% accuracy overall, as well as very good performance by class, i.e. precision and recall values above 0.99. In addition, only two types of queries have been used by Lee et al. (2005), 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 Rose & Levinson (2004) and as transactional by Broder (2002).

Another approach that led to a relatively high performance is Mendoza & Zamora (2009), 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. Our approach used one three-class model which outperforms each of the three 2-class models.

The majority of the previous approaches (Liu et al., 2006; Baeza-Yates et al., 2006; Jansen et al., 2008; Herrera et al., 2010; González-Caro & Baeza-Yates, 2011; Hernández et al., 2012; Tsukuda et al., 2013; Figueroa, 2015) obtained better classification results for the informational queries compared with navigational and transactional ones, leading to two different approaches to this problem: (a) eliminating the transactional category (Lee et al., 2005; Ashkan et al., 2009; Tsukuda et al., 2013); (b) merging some categories, e.g. informational with transactional (Liu et al., 2006), navigational with transactional (Baeza-Yates et al., 2006; González-Caro & Baeza-Yates, 2011). Some found the transactional ones more difficult to identify than the navigational ones (Figueroa, 2015), while others found the opposite (Hernández et al., 2012).

Without the domain-specific syntactic categories (i.e. levels 3, 4, 5 and L3), our results had the same tendency as the ones in Figueroa (2015), 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 Hernández et al. (2012), 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, our approach 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

Table 16: Previous approaches performance [Algorithms (Alg), Accuracy (Acc), Precision (P), Recall (R)]

| Reference | Alg | Acc | F-score | | | | P | R | Notes |
|---|---|---|---|---|---|---|---|---|---|
| Lee et al. (2005) | LR | 90% | | | | | | | 2 classes: informational and navigational |
| Liu et al. (2006) | DR | 80% | 0.81 | | | | 81.49 | 81.54 | 2 classes: C1=informational and transactional, C2=navigational |
| | | | C1 | C2 | | C1 | 73.74 | 72.84 | |
| | | | 0.73 | 0.85 | | C2 | 85.62 | 86.18 | |
| Baeza-Yates et al. (2006) | SVM | | | | | C1 | 0.7 | 0.9 | 3 classes: C1=informational, C2=non-informational (navigational and transactional), C3=ambiguous |
| | | | | | | C2 | 0.55 | 0.4 | |
| | | | | | | C3 | 0.35 | 0.2 | |
| Jansen et al. (2008) | rules | 74% | | | | | | | most errors are from misclassifying navigational and transactional queries as informational |
| Ashkan et al. (2009) | SVM | 84.5% | | | | C1 | 0.86 | 0.87 | 2 classes: C1=navigational and C2=informational |
| | | | | | | C2 | 0.81 | 0.80 | |
| Mendoza & Zamora (2009) | SVM | | 91–94% | | | | | | three 2-class models: informational/other; navigational/other; transactional/other; |
| Kathuria et al. (2010) | k-means | 94% | | | | | | | 8 clusters: 6 navigational; 1 transactional and 1 navigational |
| Herrera et al. (2010) | SVM | | 94.87 | | | | 94.87 | 94.87 | 2 classes: navigational, informational |
| | SVM | | 79.18 | | | | 79.18 | 79.18 | 3 classes: navigational, informational, transactional |
| González-Caro & Baeza-Yates (2011) | SVM | | 0.4594 | | | | 0.8238 | 0.4463 | 2 classes: C1=informational and C2=non-informational (transactional and navigational) |
| | | | C1 | C2 | | C1 | 0.7227 | 0.9915 | |
| | | | 0.82 | 0.68 | | C2 | 0.8917 | 0.2948 | |
| Hernández et al. (2012) | NB | | C1 | C2 | C3 | C1 | 0.929 | 0.886 | 3 classes: C1=informational, C2=transactional, C3=navigational |
| | | | 0.86 | 0.82 | 0.39 | C2 | 0.84 | 0.810 | |
| | | | | | | 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 | |
| Tsukuda et al. (2013) | SVM | 64.4% | | | | | | | 2 classes: navigational and informational |
| Figueroa (2015) | MaxEnt | 82.22% | | | | C1 | | 88.23 | 3 classes: C1= informational, C2=navigational, C3=resource/transactional |
| | | | | | | C2 | | 79.42 | |
| | | | | | | 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 | |

number of features, and an automatic grammar-based approach for transforming queries into the syntactic patterns representation.

## 9. Conclusions and future work

In this paper we proposed the Customised Grammar Framework (CGF) for the automatic classification of text through machine learning by taking advantage of domain-specific information and by preserving the structure of 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 we proposed a formal grammar-based approach.

We applied the framework to the query classification problem, and our results indicate that our approach outperforms previous ones, both overall, as well as for each type of query. In addition, our approach addresses one of the major issues in text representation, i.e. large sparse datasets, by requiring a significantly smaller number of features. While our framework was tested on query classification, the proposed approach can be applied to other text classification problems; we will investigate this in future work.

In addition, one of the limitations that affected the performance of our approach that we aim to investigate in future work is the problem of class imbalance as query datasets suffer from class imbalance between the labels; this problem affects the classification results, so applying different imbalance algorithms e.g (cost-sensitive and SMOTE) may lead to the improvement of query classification.

## References

## References

Altrabsheh, N., Cocea, M., & Fallahkhair, S. (2014). 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.

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

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

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

Beitzel, S. M., Jensen, E. C., Frieder, O., Grossman, D., Lewis, D. D., Chowdhury, A., & Kolcz, A. (2005). 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.

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

Broder, A. (2002). A taxonomy of web search. In *ACM Sigir forum* (pp. 3–10). ACM volume 36.

Calderón-Benavides, L., González-Caro, C., & Baeza-Yates, R. (2010). Towards a deeper understanding of the users query intent. In *SIGIR 2010 Workshop on Query Representation and Understanding* (pp. 21–24).

Chomsky, N. (1993). *Lectures on government and binding: The Pisa lectures*. 9. Walter de Gruyter.

Conneau, A., Schwenk, H., Barrault, L., & Lecun, Y. (2017). 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* (pp. 1107–1116). volume 1.

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

Gayo-Avello, D. (2009). A survey on session detection methods in query logs and a proposal for future evaluation. *Information Sciences*, *179*, 1822 – 1843. URL: http://www.sciencedirect.com/science/article/pii/S002002550900053X. doi:http://dx.doi.org/10.1016/j.ins.2009.01.026. Special Section: Web Search.

Ghose, A., & Ipeirotis, P. G. (2011). Estimating the helpfulness and economic impact of product reviews: Mining text and reviewer characteristics. *IEEE Transactions on Knowledge and Data Engineering*, *23*, 1498–1512. doi:10.1109/TKDE.2010.188.

Glorot, X., Bordes, A., & Bengio, Y. (2011). 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).

Gong, Z., & Yu, T. (2010). 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.

González-Caro, C., & Baeza-Yates, R. (2011). A multi-faceted approach to query intent classification. In R. Grossi, F. Sebastiani, & F. Silvestri (Eds.), *String Processing and Information Retrieval: 18th International Symposium, SPIRE 2011, Pisa, Italy, October 17-21, 2011. Proceedings* (pp. 368–379). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-642-24583-1_36.

Han, H.-q., Zhu, D.-H., & Wang, X.-f. (2009). 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.

Hardy, H., & Cheah, Y.-N. (2013). Question classification using extreme learning machine on semantic features. *Journal of ICT Research and Applications*, *7*, 36–58.

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

Herrera, M. R., de Moura, E. S., Cristo, M., Silva, T. P., & da Silva, A. S. (2010). Exploring features for the automatic identification of user goals in web search. *Information Processing & Management*, *46*, 131 – 142. URL: `http://www.sciencedirect.com/science/article/pii/S0306457309001058`. doi:`http://dx.doi.org/10.1016/j.ipm.2009.09.003`.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*, 1735–1780.

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

Iyyer, M., Manjunatha, V., Boyd-Graber, J., & Daumé III, H. (2015). Deep unordered composition rivals syntactic methods for text classification. 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)* (pp. 1681–1691). volume 1.

Jansen, B. J., & Booth, D. (2010). Classifying web queries by topic and user intent. In *CHI'10 Extended Abstracts on Human Factors in Computing Systems* (pp. 4285–4290). ACM.

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

Jiang, D., Leung, K. W.-T., & Ng, W. (2016). Query intent mining with multiple dimensions of web search data. *World Wide Web*, *19*, 475–497. URL: `http://dx.doi.org/10.1007/s11280-015-0336-2`. doi:`10.1007/s11280-015-0336-2`.

Jung, S.-W., & Kwon, H.-C. (2006). A scalable hybrid approach for extracting head components from web tables. *IEEE Transactions on Knowledge and Data Engineering*, *18*, 174–187. doi:`10.1109/TKDE.2006.19`.

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

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

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

King, M. (1983). *Parsing natural language*. Academic Press London.

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

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

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

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

Li, Y., Su, L., Chen, J., & Yuan, L. (2017). Semi-supervised learning for question classification in cqa. *Natural Computing*, *16*, 567–577.

Liu, B., Li, X., Lee, W. S., & Yu, P. S. (2004). Text classification by labeling words. In *AAAI* (pp. 425–430). volume 4.

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

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

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

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

Mohasseb, A., Bader-El-Den, M., & Cocea, M. (2018). Question categorization and classification using grammar based approach. *Information processing and management*, .

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

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

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

Morrison, J. B., Pirolli, P., & Card, S. K. (2001). 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.

Muhammad, A., Wiratunga, N., & Lothian, R. (2015). Context-aware sentiment analysis of social media. In M. M. Gaber, M. Cocea, N. Wiratunga, & A. Goker (Eds.), *Advances in Social Media Analysis* (pp. 87–104). Cham: Springer International Publishing. URL: `https://doi.org/10.1007/978-3-319-18458-6_5`. doi:`10.1007/978-3-319-18458-6_5`.

Nijholt, A. (1980). *Context-free grammars: covers, normal forms, and parsing*. 93. Springer Science & Business Media.

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

Pass, G., Chowdhury, A., & Torgeson, C. (2006). A picture of search. In *InfoScale - Proceedings of the 1st international conference on Scalable information systems*. ACM Press, New York. doi:`10.1145/1146847.1146848`.

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

Peters, R. A. (1968). *A linguistic history of English*. Houghton Mifflin.

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

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

Shi, Y., Yao, K., Tian, L., & Jiang, D. (2016). Deep lstm based feature mapping for query classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 1501–1511).

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

Sushmita, S., Piwowarski, B., & Lalmas, M. (2010). Dynamics of genre and domain intents. In P.-J. Cheng, M.-Y. Kan, W. Lam, & P. Nakov (Eds.), *Information Retrieval Technology: 6th Asia Information Retrieval Societies Conference, AIRS 2010, Taipei, Taiwan, December 1-3, 2010. Proceedings* (pp. 399–409). Berlin, Heidelberg: Springer Berlin Heidelberg.

Taboada, M., Brooke, J., Tofiloski, M., Voll, K., & Stede, M. (2011). Lexicon-based methods for sentiment analysis. *Computational linguistics*, *37*, 267–307.

Tang, B., He, H., Baggenstoss, P. M., & Kay, S. (2016). A bayesian classification approach using class-specific features for text categorization. *IEEE Transactions on Knowledge and Data Engineering*, *28*, 1602–1606. doi:10.1109/TKDE.2016.2522427.

Tsukuda, K., Sakai, T., Dou, Z., & Tanaka, K. (2013). Estimating intent types for search result diversification. In R. E. Banchs, F. Silvestri, T.-Y. Liu, M. Zhang, S. Gao, & J. Lang (Eds.), *Information Retrieval Technology: 9th Asia Information Retrieval Societies Conference, AIRS 2013, Singapore, December 9-11, 2013. Proceedings* (pp. 25–37). Berlin, Heidelberg: Springer Berlin Heidelberg. URL: http://dx.doi.org/10.1007/978-3-642-45068-6_3. doi:10.1007/978-3-642-45068-6_3.

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

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

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

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

Wang, P., Xu, J., Xu, B., Liu, C., Zhang, H., Wang, F., & Hao, H. (2015). 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)* (pp. 352–357). volume 2.

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

Wu, D., Zhang, Y., Zhao, S., & Liu, T. (2010). 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.

Yang, H., Hu, Q., & He, L. (2015). Learning topic-oriented word embedding for query classification. In T. Cao, E.-P. Lim, Z.-H. Zhou, T.-B. Ho, D. Cheung, & H. Motoda (Eds.), *Advances in Knowledge Discovery and Data Mining: 19th Pacific-Asia Conference, PAKDD 2015, Ho Chi Minh City, Vietnam, May 19–22, 2015, Proceedings, Part I* (pp. 188–198). Cham: Springer International Publishing.

Yang, K., Cai, Y., Huang, D., Li, J., Zhou, Z., & Lei, X. (2017). 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.

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

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

Zhang, D., & Lee, W. S. (2003). 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.

Zhang, L., Wang, S., & Liu, B. (2018a). Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *8*, e1253.

Zhang, Q., Yang, L. T., Chen, Z., & Li, P. (2018b). A survey on deep learning for big data. *Information Fusion*, *42*, 146–157.

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

## Appendix: 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 | $QW_{Which}$ |