Noname manuscript No.
(will be inserted by the editor)

From symbolic to sub-symbolic information in question classification

João Silva · Luísa Coheur · Ana Cristina Mendes · Andreas Wichert

Received: date / Accepted: date

Abstract Question Answering (QA) is undoubtedly a growing field of current research in Artificial Intelligence. Question classification, a QA subtask, aims to associate a category to each question, typically representing the semantic class of its answer. This step is of major importance in the QA process, since it is the basis of several key decisions. For instance, classification helps reducing the number of possible answer candidates, as only answers matching the question category should be taken into account. This paper presents and evaluates a rule-based question classifier that partially founds its performance in the detection of the question headword and in its mapping into the target category through the use of WordNet. Moreover, we use the rule-based classifier as a features' provider of a machine learning-based question classifier. A detailed analysis of the rule-base contribution is presented. Despite using a very compact feature space, state of the art results are obtained.

1 Introduction

In question answering, the goal of question classification is to map a question into a category that represents the type of the information that is expected to be present in the final answer. Question classification is a very important step in the question answering process, as the selected question category can be used for several different purposes. First, it can help narrowing down the number of possible answer candidates. For example, knowing that a question belongs to the category City, allows to only consider cities as possible answers. Second, depending on the question category, different strategies can be chosen to find an answer. For instance, if a question is assigned to the category Definition, possible answers can be searched in encyclopedic sources –

João Silva · Luísa Coheur · Ana Cristina Mendes · Andreas Wichert
INESC-ID Lisboa/IST
Av. Prof. Cavaco Silva
2780-990 Porto Salvo Tagus Park, Portugal
Tel.: +351-214 233 577
Fax: +351-214 233 252
E-mail: {joao.silva,luisa.coheur,ana.mendes}@l2f.inesc-id.pt
E-mail: {andreas.wichert}@tagus.ist.utl.pt

such as Wikipedia –, which are more likely to contain a suitable answer. Furthermore, a misclassified question can hinder the ability to reach a correct answer, because it can lead to wrong assumptions about the question. Hence, in order to obtain good performance, it is of crucial importance for any question answering system to have an accurate question classifier (for a detailed analysis on the impact of question classification in question answering see (Moldovan et al., 2003)).

In this paper we present a stand-alone rule-based question classifier, that couples two methods to obtain its results:

- a direct (pattern) match is performed for specific questions. For instance, *Who is Mozart?* is directly mapped into HUMAN:DESCRIPTION;
- headwords are identified (by a rule-based parser) and mapped into the question classification (by using WordNet (Fellbaum, 1998)). For example, in the question *What is Australia's national flower?* the headword *flower* is identified and mapped into the category ENTITY:PLANT.

Then, we show how the rule-based question classifier can be enhanced in a machine learning environment. Several experiments are carried out, and the information the rule-based question classifier manipulates and generates is used as features, as well as merged with other features within a Support Vector Machine (SVM). Our strategy results in an improvement of about 8% over the stand-alone rule-based classifier and state of the art results in question classification are obtained when compared with previous work over the same corpora, although a smaller feature space is used.

Our approach also follows in the old Artificial Intelligence discussion. On the one hand, symbolic information is generally characterized by static tokens (symbols) used to denote or refer to something other than themselves, namely other entities in the world (Tarski, 1956). In this context, symbols alone do not represent any utilizable knowledge as, for example, they cannot be used for a definition of similarity criteria between themselves. However, systems operating with symbolic information have the advantage of dealing with hard-coded, explicit rules (Simon, 1969; Newell) which allow to represent complex relationships and can be easily manipulated. In this perspective, our rule-based classifier is a symbolic system and the presented rules are an important resource, as besides the purpose of question classification, they can be used in other natural language applications, as for instance query expansion within a question answering system.

On the other hand, systems operating with sub-symbolic information fall into the learning paradigm and the information they control is difficult to manipulate externally. Nevertheless, sub-symbolic information represent properties of world entities, allowing similarity between entities to be defined as a function of the features they have in common (Sun, 1995). When the information used by the rule-based classifier is used as features by the SVM, we are dealing with the sub-symbolic level. Therefore, in this paper we propose to enhance a symbolic system within a sub-symbolic environment.

The reminder of the paper is organized as follows. Section 2 describes related work and positions our work. Section 3 presents a general overview of the rule-based classifier. Sections 4 and 5 present the rule-based classifier, detailing the headword extraction and classification processes. Empirical results and their analysis are presented in Section 6. Conclusions and future directions are presented in Section 7.

All the employed rules and software used in this work can be downloaded from `http://qa.l2f.inesc-id.pt/wiki/index.php/Resources`.

2 Related Work

2.1 Question Type Taxonomy

The set of question categories into which the questions are to be assigned is referred to as the question type taxonomy. As an example, the question *What country borders Portugal?* ought to be classified into the category COUNTRY. Several question type taxonomies have been proposed in the literature (Hermjakob et al., 2002; Moldovan et al., 2000; Li and Roth, 2002), some of which are hierarchical, some are flat. For instance, a popular taxonomy is the one used in Webclopedia (Hermjakob et al., 2002), a web-based question answering system, that uses a hierarchical taxonomy spanning over one hundred and eighty categories. To date, it is probably the broadest taxonomy proposed for question classification. Nevertheless, one of the most widely known taxonomies for question classification is Li and Roth's two-layer taxonomy (Li and Roth, 2002), which consists of a set of six coarse-grained categories and fifty fined-grained ones, as shown in Table 1. This taxonomy is widely used in the machine learning community (Li and Roth, 2002; Blunsom et al., 2006; Huang et al., 2008; Zhang and Lee, 2003), probably due to the fact that the authors have published a set of nearly 6,000 labeled questions, the University of Illinois at Urbana-Champaign dataset – from now on the UIUC dataset – that is freely available on the Web, making it a very valuable resource for training and testing machine learning models.

Table 1 Li and Roth's two-layer taxonomy for question classification.

Coarse	**Fine**
ABBREVIATION	abbreviation, expansion
DESCRIPTION	definition, description, manner, reason
ENTITY	animal, body, color, creative, currency, medical disease, event, food, instrument, language, letter, other, plant, product, religion, sport, substance, symbol, technique, term, vehicle, word
HUMAN	description, group, individual, title
LOCATION	city, country, mountain, other, state
NUMERIC	code, count, date, distance, money, order, other, percent, period, speed, temperature, size, weight

As all the different question classifiers that follow Li and Roth's taxonomy and are tested in the UIUC dataset can be straightforward compared, we also adopt Li and Roth's taxonomy, and test and evaluate our experiments in the UIUC dataset.

2.2 Rule-based Question Classifiers

Most of the early approaches to question classification followed a rule-based strategy and used manually built classification rules to determine the question type – for instance, MULDER (Kwok et al., 2001), determined the question type just by looking to the question's interrogative pronoun. Nowadays, the rule-based strategy is still followed

(Amaral et al., 2008; Mendes et al., 2008) and even machine learning-based question classifiers make use of rule-based ones, as features suppliers. For instance, in (Huang et al., 2008) a mini-rule-based question classifier (not evaluated) is used to provide semantic features to a machine learning classifier.

However, in what concerns the question taxonomy typically used by the rule-based question classifiers, many question answering systems following this paradigm use their own private taxonomies, such as (Kwok et al., 2001; Saquete et al., 2009; Amaral et al., 2008). These systems usually report very precise results in this task (albeit with very little recall, since the manually built patterns cover only a small set of questions). Nevertheless, the differences between their taxonomies make those systems' classification processes impossible to compare.

In this paper we present a rule-based question classifier, that follows recent tendencies in machine learning question classification. As in (Huang et al., 2008), we also perform headword extraction and use WordNet to map the obtained headwords on the question categories, although by following different strategies. In fact, we take a full advantage of this process, which results in a stand-alone rule-based question classifier.

2.3 Machine Learning-based Question Classifiers

Many supervised machine learning approaches to question classification have been devised over the last few years (Li and Roth, 2002; Blunsom et al., 2006; Huang et al., 2008; Metzler and Croft, 2005). These approaches vary according to the classifier in use – for instance (Pan et al., 2008) and (Metzler and Croft, 2005) use SVM, and (Blunsom et al., 2006) approach uses Log-linear Models – but mainly to the features feeding the classifier. However, these systems have something in common: they are all trained and tested on the UIUC dataset, leading to a straightforward comparison between them. In the following, unless otherwise stated, all the presented results outcome from the same dataset.

Machine learning approaches – even simple approaches using just surface text features like bag-of-words – tend to achieve a high accuracy for the question classification task as long as a corpus of labeled question is available. For instance, in the experiments carried out by (Zhang and Lee, 2003), a Naïve Bayesian classifier was used for the task of question classification. Experimental results showed that just with bag-of-ngrams features, Naïve Bayesian classifiers trained on 5,500 examples of labeled questions, present an accuracy of 83.2% under the coarse grained category. However, the results also indicated that in order to achieve this accuracy, the training sets needed to be quite large, since when the classifier was trained with just 1,000 examples, the accuracy dropped to 53.8%. These authors also compared several machine learning techniques applied to the problem of question classification, with the experimental results showing that, using only surface text features, SVM significantly outperforms every other technique, such as Naïve Bayes and Sparse Network of Winnows (SNoW) learning architecture[1] (Carlson et al., 1999). Moreover, (Zhang and Lee, 2003) also proposed a tree Kernel, which measures the similarity between two syntactic trees, by counting the number of common tree fragments. With the tree kernel, the authors achieved 90.0% accuracy for coarse-grained classification.

[1] Available at `http://l2r.cs.uiuc.edu/~danr/snow.html` (last accessed in June 2010).

Further improvements of the tree kernel proposed by (Zhang and Lee, 2003) are described in (Pan et al., 2008), where semantic features (resulting from a set of 20 semantic classes such as PERSON, LOCATION and CREATURE) are incorporated into the kernel. The experimental results evidenced that these semantic classes just by themselves are very helpful to question classification, resulting in an accuracy of 93.2%. Moreover, these semantic classes can also be used to augment the training set, as demonstrated by (Bhagat et al., 2005). With all semantic features combined, these authors achieved an accuracy of 94.0% which, to date, outperforms every other question classifier on the standard training set of Li and Roth, for coarse-grained classification.

A hierarchical question classifier based on the SNoW learning architecture was developed in (Li and Roth, 2002). The hierarchical classifier was composed of two simple SNoW classifiers with the Winnow update rule, where the first classifier was used to classify a question into coarse grained categories, and the second into fine grained ones. A feature extractor was used to automatically extract primitive features from each question, ranging from syntactic to semantic features. Also, a set of complex features is composed over the primitive features. The reported results achieved an accuracy of 91.0% and 84.2%, for coarse and fine grained classification, respectively. Results showed that the semantically related words are the most prominent feature, and that the use of a hierarchical classifier provides no significant advantages over a flat classifier. Therefore, in (Li et al., 2008), the authors went on to further improve their former SNoW based question classifier, by including even more semantic features, such as WordNet senses and additional named entities. However, only a slight accuracy improvement of roughly 1% was attained, over the coarse and fine grained categories.

The notions of *informer span* and *headword* were used in (Krishnan et al., 2005) and (Huang et al., 2008; Metzler and Croft, 2005), respectively, as features to train a SVM. Authors theorized that a question can be accurately classified using very few words, which correspond to the object of the question. The main difference between these works is that in the informer span can consider a sequence of words, while the headword is only a single word. For example, in the question *What is the capital of Italy?*, the informer span is *capital of Italy* and the headword is *capital*. In addition, (Krishnan et al., 2005) (and also (Metzler and Croft, 2005)) enriches the feature space with all hypernyms from all senses of the informer span (headword), while (Huang et al., 2008) uses all hypernyms of the headword at depth ≤ 6. All these authors evaluated their experiments using the UIUC dataset, with (Krishnan et al., 2005) reaching an accuracy of 93.4% and 86.2% for coarse- and fine-grained classification, respectively, while (Huang et al., 2008) attained 93.4% and 89.2%, and (Metzler and Croft, 2005) 90.2% and 83.6%.

In this paper we will use the rule-based question classifier as a features' provider for a SVM.

3 Rule-based question classification: overall architecture

The rule-based classifier that we present in this work starts by triggering a set of 60 manually built patterns, that are matched against each question. If the match is successful, a category is returned and the question is classified; otherwise the classifier searches for the question headword and extracts it. Then, the headword hypernyms are followed until one is associated with a possible question category. For instance, the manually built patterns are able to correctly classify the sentence *When did Hawaii* *become a state ?* with the fine-grained category NUMERIC:DATE. However, no pattern matches the question *What person 's head is on a dime?*. Therefore, its headword – *person* – is (correctly) identified. By following its hypernyms, the classifier correctly tags it as HUMAN:INDIVIDUAL. Algorithm 1 summarizes the entire process of question classification.

Algorithm 1 Rule-based question classification algorithm

```
procedure CLASSIFY(question)
    if PATTERN-MATCHES?(question) then
        return category                    ▷ Returns the question category (Table 2)
    else
        headword ← EXTRACT-QUESTION-HEADWORD(question.tree, rules)          ▷
Algorithm 2
        return HEADWORD-CATEGORY(headword, groups)                ▷ Algorithm 3
    end if
end procedure
```

Thus, we start by compiling the set of manually built patterns – some of which are adapted from (Huang et al., 2008). A simplified version of the patterns is presented in Table 2.

These rules cover all the questions that begin with the *Wh*-word *Who*, *Where* or *When*, as its *Wh*-word itself represents the information that is missing. However, there are still some (albeit few) other question categories that can be recovered by this direct pattern matching as, in these questions, the definition of headword would not help classification. For instance, in DESCRIPTION:DEFINITION questions such as *What is a bird?*, the headword *bird* is futile because the question is asking for a definition. Moreover, it can mislead the classifier into assigning the category ENTITY:ANIMAL to it.

In the following sections we detail the headword extraction process, as well as the mapping between the recovered headwords and the question category.

4 Headword Extraction

In the literature, there is no agreement on what exactly a question headword is, and it is out of the scope of this paper to discuss this concept. Our main interest is to find the word that will lead us to the correct categorization of the question. Therefore, we can say that the definition followed in this work is similar to the one presented in (Huang et al., 2008): the headword of a given question is a word that represents the object that is being sought after. In the following, we present in bold face some examples of questions' headwords:

(1) *What is Australia's national **flower**?*
(2) *Name an American made **motorcycle**.*
(3) *Which **country** are Godiva chocolates from?*
(4) *What is the name of the highest **mountain** in Africa?*

In Example 1, the headword *flower* provides the classifier with an important clue to correctly classify the question to ENTITY:PLANT. By the same token, *motorcycle* in

Category	**Question pattern description**	**Example**
ABBREV.:EXPANSION	begins with *What do(es)* and ends with an acronym – i.e., a sequence of capital letters possibly intervened by dots –, followed by *stands for/mean*; begins with *What is/are* and ends with an acronym	*What does AIDS mean?* *What is F.B.I.?*
DESCRIPTION:DEF.	begins with *What is/are* and is followed by an optional determiner and a sequence of nouns	*What is ethology?*
ENTITY:TERM	begins with *What do you call*	*What do you call...*
ENTITY:SUBSTANCE	begins with *What is/are* and ends with *composed/made of*	*What is glass made of?*
DESCRIPTION:REASON	begins with *What causes*	*What causes asthma?*
HUMAN:DESCRIPTION	begins with *Who is/was* and is followed by a proper noun	*Who was Mozart?*

Table 2 A set of question patterns used to avoid extracting a headword, when not needed.

Example 2 renders hints that help classify the question to ENTITY:VEHICLE. Indeed, all of the aforementioned examples' headword serve as an important clue to unveil the question's category, which is why we dedicate a great effort to its accurate extraction.

4.1 Headword Extraction Algorithm and Head Rules

In (Metzler and Croft, 2005) is described a method to perform headword extraction. The authors find the first noun phrase and extract the rightmost word tagged as a noun. Although they report an accuracy of approximately 90%, we decided to use a rule-based method, allowing a more precise headword extraction, that we explain in the following. As it will be seen in Section 6 our method will attain an accuracy of 96.9% for coarse-grained categories.

Our approach for the extraction of the question headword requires a parse tree of the question. We use the Berkeley Parser (Petrov and Klein, 2007) for the purpose of parsing questions, trained on the QuestionBank (Judge et al., 2006), a treebank of 4,000 parse-annotated questions. Figure 1 shows the parse tree of Example 1, where punctuation is omitted for the sake of simplicity.

The resulting parse tree of a question is then traversed top-down to find the question headword, using Algorithm 2. Considering a non-terminal X with production rule

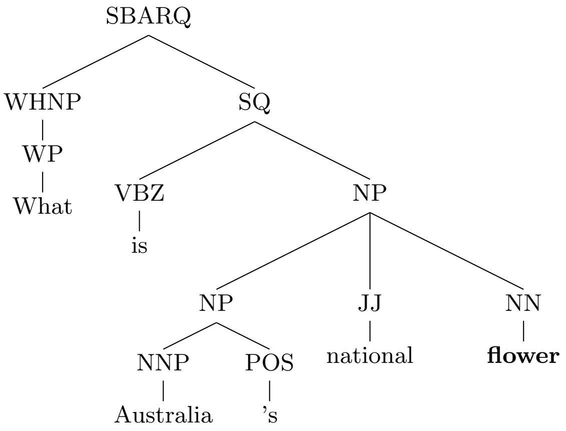

Fig. 1 Parse tree of the question *What is Australia's national flower?*. The question headword is in bold face.

$X \to Y_1 \cdots Y_n$, the algorithm uses a pre-defined set of rules – the head-rules – in order to decide which of the $Y_1 \cdots Y_n$ is the head or contains it (Apply-Rules). This process is then repeated recursively from Y_i, until a terminal node is reached.

Algorithm 2 Question headword extraction algorithm

```
procedure Extract-Question-Headword(tree, rules)
    if Terminal?(tree) then
        return tree                                        ▷ Returns the headword
    else
        child ← Apply-Rules(tree, rules)      ▷ Determine which child of tree is the head
        return Extract-Question-Headword(child, rules)
    end if
end procedure
```

The head-rules used in this work are a heavily modified version of those given in (Collins, 1999), specifically tailored to extract headwords from questions. A subset of these rules is listed in Table 3.

As an example of how the algorithm works, consider the parse tree of Figure 1. We start by running the root production rule *SBARQ* → *WHNP SQ* through the head-rules, which specify that the search is conducted from left to right, to find the first child which is an *SQ*, *S*, *SINV*, *SBARQ*, or *FRAG*, thereby identifying *SQ*. A similar reasoning can be applied to determine *NP* as the head of *SQ*. Having the production rule *NP* → *NP JJ NN*, we search from right to left *by position*[2], i.e., starting with the child at the right-most position, we first test it against *NP* – which does not match –

[2] Note that if the direction of *NP* was *Right* by category instead of by position, the left *NP* would be chosen, as we would have searched first by category and then by position, i.e., we would have checked first if there was any *NP* in the entire right-hand-side of the rule, rather than testing it against the child at the right-most position.

Parent	**Direction**	**Priority List**
S	Left	VP S FRAG SBAR ADJP
SBARQ	Left	SQ S SINV SBARQ FRAG
SQ	Left	NP VP SQ
NP	Right *by position*	NP NN NNP NNPS NNS NX
PP	Left	WHNP NP WHADVP SBAR
WHNP	Left	NP
WHPP	Right	WHNP WHADVP NP SBAR

Table 3 Subset of the head-rules used to determine the question headword, also known as a head percolation table. *Parent* is the non-terminal on the left-hand-side of a production rule. *Direction* specifies whether to search from the left or right end of the rule, either by category first and then by position (by default), or vice-versa when explicitly mentioned. *Priority List* presents the right-hand-side categories ordered by priorities, with the highest priority on the left.

and then against *NN*, which yields a match. At last, since *NN* is a pre-terminal, the terminal *flower* is returned as the headword.

4.2 Non-trivial Head Rules

There are, however, a few exceptions to the head-rules. Consider the parse tree of Example 3 (*Which* ***country*** *are Godiva chocolates from?*), depicted in Figure 2.

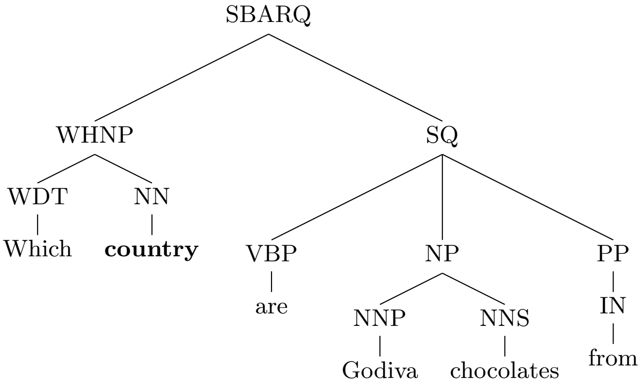

Fig. 2 Example of a parse tree that requires the use of a non-trivial rule for *SBARQ*.

In this example, if we follow the aforesaid head-rules, *chocolates* would be extracted as the headword, instead of *country* which is the correct headword. This happens because the rule for *SBARQ* chooses *SQ* over *WHNP*, where *country* lies in. A possible solution would to be to prioritize *WHNP* over *SBARQ*, but this would cause wrong headwords to be extracted in other examples, such as in the parse tree of Figure 1, which would return *What*.

Clearly, this problem cannot be solved by modifying the head percolation table, which is why we created a set of what we shall refer to as *non-trivial rules*, which determine the correct headword for situations that cannot be covered by the head percolation table. These rules are described in the following.

- Considering that *WHXP* refers to a *Wh*-phrase – *WHNP*, *WHPP*, *WHADJP* or *WHADVP* – the first rule states that when *SBARQ* has a *WHXP* child with at least two children, *WHXP* is returned. This situation occurs in Figure 2.
- After applying the first rule, if the resulting head is a *WHNP* containing an *NP* that ends with a possessive pronoun (*POS*), we return the *NP* as the head. Without this second rule, the headword for the example in Figure 3 would be *capital* instead of *country*. Note that this rule only applies inside a *WHNP*, allowing the algorithm to correctly extract *birthday* as the headword of the question *What* $[_{\mathrm{SQ}}$ *is* $[_{\mathrm{NP}}$ *Martin Luther King* $[_{\mathrm{POS}}$ *'s*]] $[_{\mathrm{NN}}$ ***birthday***]]*?* using the head-rules.

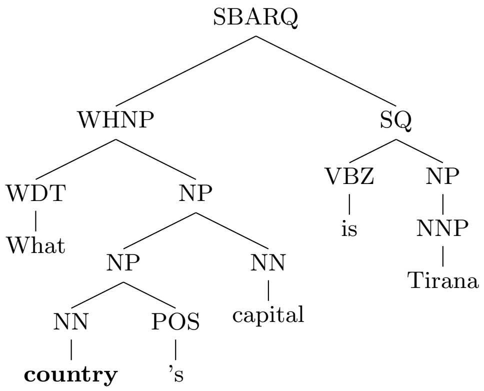

Fig. 3 Example of a parse tree that requires two non-trivial rules, the first for *SBARQ* and the second for *WHNP*

.

- However, there is still another particular situation that needs to be considered, as there are some headwords, such as *type* and *kind*, which serve merely as a *reference* to the real headword. Figure 4 illustrate this situation. For instance, the extracted headword of Figure 4 is *kind*, which is referring to a kind of *animal*. In this situation, *animal* would be a much more informative headword. To fix this problem, if the extracted headword is either *name*, *kind*, *type*, *part*, *genre* or *group*, and its sibling node is a prepositional phrase *PP*, we use *PP* as the head and proceed with the algorithm. Note that this rule does not apply if there is not a sibling *PP*, as in *What is Mao's second name?*.

Algorithm 2 still works in the same manner and the only exception is that the Apply-Rules procedure now takes the non-trivial rules into account to determine the head of a non-terminal node.

5 Mapping Headwords into question categories

In this section we explain how the question category is obtained from the headword, by using WordNet (Fellbaum, 1998). Consider the following example questions, with their corresponding headword in bold face:

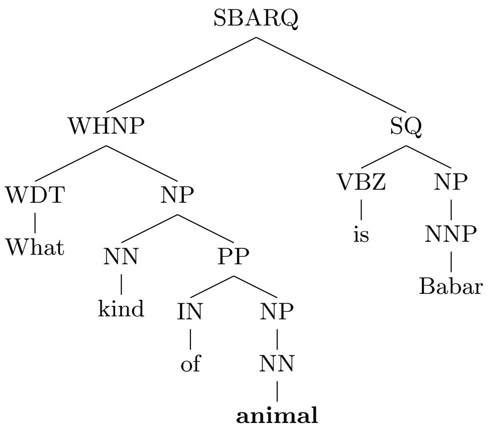

Fig. 4 Examples of a parse tree that require a non trivial rule in order not to return *Kind* as headword.

(5) *What* ***explorer*** *was nicknamed Iberia's Pilot?*
(6) *What* **actor** *first portrayed James Bond?*
(7) *What* **dictator** *has the nickname "El Maximo"?*

Even though all of the above examples fall under the HUMAN:INDIVIDUAL category, the question headword is different in all of them, which limits the usefulness of the headword in the question classification process. These headwords do, however, share a common trait: they are all subordinates (hyponyms) of the word *person*, that is to say, they are all more specific senses of *person* – the superordinate (hypernym). Knowing this information would be useful to accurately classify the previous questions.

We exploited this observation by using WordNet's lexical hierarchy to associate the headword with a higher-level semantic concept, which represents a question category. This was accomplished as follows. First, sets of related WordNet synsets were manually grouped together into fifty clusters, each representing a question category. Some of these clusters are shown in Table 4.

Secondly, given a question headword, we map it into a WordNet synset using a set of heuristics, which we will describe later. Finally, and since WordNet can be seen as a directed acyclic graph, with synsets as vertices and lexical relations – such as hypernym – as edges, we employ a breadth-first search on the translated synset's hypernym tree, in order to find a synset that pertains to any of the pre-defined clusters. Algorithm 3 details this process.

As an illustration of how the algorithm works, consider the headword *actor* of Example 6 and its hypernym tree depicted in Figure 5. The first three levels of the tree yield no results, since no cluster contains any of the synsets at these levels. At the fourth level of the hypernym chain, however, a member of the HUMAN:INDIVIDUAL cluster is found – *person* –, and thereby the algorithm terminates, returning the associated cluster, that is, HUMAN:INDIVIDUAL.

In order for this process to be effective, it is crucial to translate the headword into the appropriate WordNet synset. This, however, is not a trivial undertaking due, in grand part, to homonymous and polysemous words – i.e., words with the same spelling but different meanings –, which have multiple senses (synsets) in WordNet.

Category (Cluster name)	Synsets	Example hyponyms
ENTITY:ANIMAL	animal, animate_being, beast, brute, creature, fauna	mammal, fish, cat
	animal_group	flock, herd, breed
ENTITY:CREATIVE	show	movie, film, tv show
	music	song, tune, hymn
	writing, written material, piece of writing	book, poem, novel
ENTITY:PLANT	vegetation, flora, botany	forest, garden
	plant, flora, plant life	flower, tree, shrub
HUMAN:INDIVIDUAL	person, individual, someone, somebody, mortal	actor, leader
	spiritual being, supernatural being	god, angel, spirit
	homo, man, human being, human	homo sapiens
NUMERIC:DISTANCE	distance	altitude, elevation
	dimension	width, length, height

Table 4 Examples of clusters that aggregate similar synsets together.

Algorithm 3 Headword category extraction algorithm

```
procedure HEADWORD-CATEGORY(headword, groups : associative array synset → group)
    root-synset ← MAP-TO-SYNSET(headword)
    ENQUEUE(root-synset, queue)
    while ¬EMPTY?(queue) do
        synset ← DEQUEUE(queue)
        if synset is in groups then
            return groups[synset]
        else
            for each hypernym in DIRECT-HYPERNYMS-FOR(synset) do
                ENQUEUE(hypernym, queue)
            end for
        end if
    end while
    return not found
end procedure
```

An example of such a word is *capital* which, in WordNet[3], can refer to *wealth in the form of money* (capital#1), *a seat of government* (capital#2), or even an adjective as in *of capital importance* (capital#3). For instance, in the question *What is the **capital** of Portugal?*, the correct sense for the headword is capital#2. Failing to identify it as so

[3] WordNet includes more senses for the word *capital*; for the sake of brevity, we have selected only three senses to illustrate our examples.

```
Sense 1
actor, histrion, player, thespian, role player -- (a theatrical performer)
⇒ performer, performing artist
 ⇒ entertainer
   ⇒ person, individual, someone, somebody, mortal, soul
     ⇒ organism, being
       ⇒ living thing, animate thing
         ⇒ whole, unit
           ⇒ object, physical object
             ⇒ physical entity
               ⇒ entity
```

Fig. 5 Hypernym tree for the first sense of the synset *actor*.

can introduce noise to the classifier – e.g., selecting capital#1 could mislead the classifier into categorizing the question into NUMERIC:MONEY in lieu of LOCATION:CITY, which is the correct.

In this work, we utilize the following three heuristics to aid headword sense disambiguation:

- WordNet synsets are organized according to four part-of-speeches (POS): nouns, adjectives, verbs, and adverbs. This allows for some partial disambiguation of the question headword, by converting the headword's Penn-style POS tag into the corresponding WordNet POS, and then using it to retain only those senses that belong to the converted POS. This would eliminate the adjectival sense of *capital* in the aforesaid example.
- Besides single words, WordNet also contains entries for compound words (Sharada and Girish, 2004), which are a combination of two or more words that constitute a single unit of meaning, such as *mountain range* (Figure 6). These are very useful, because they are often monosemous and hence do not require further disambiguation. In this work, we try to form compound words from the question headword using the following two strategies[4]: (1) the headword is combined with nouns and/or adjectives to its left that act as pre-modifiers (e.g., World **Cup**); (2) the headword is combined with a prepositional phrase to its right that acts as a post-modifier (e.g. **capital** of Portugal (which happens to be in the WordNet). Another example is **pH** scale for the question *What is the pH scale?*). The resulting combination is then used, if a *valid* compound word was formed (i.e., if it exists in WordNet).
- At last, if after applying the previous two heuristics we are still left with more than one sense for the headword, we opt for the first (most frequent) WordNet sense. This decision results from several experiments where the fine-grained category classification attained better results by following only the first WordNet sense.

6 Empirical Evaluation

As previous mentioned, the empirical evaluation of the question classifier was carried out on the UIUC dataset, the publicly available data set of the Cognitive Computing

[4] Many other forms of compound words, such as hyphenated forms (e.g., anti-inflammatory) and juxtapositions (e.g., keyword), are (typically) already present in WordNet as single words.

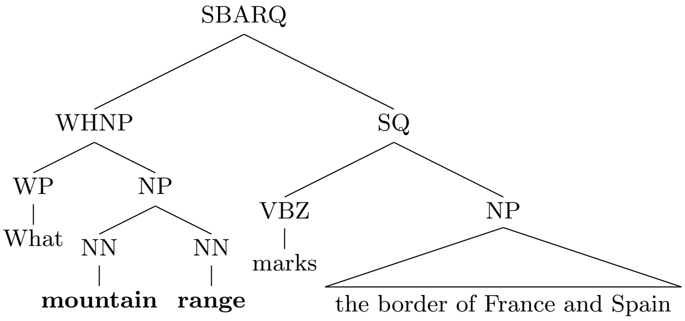

Fig. 6 Example of a question for which a compound headword is helpful. The compound headword *mountain range* can help to classify the question into LOCATION:MOUNTAIN, while *range* does not.

Group at University of Illinois at Urbana-Champaign[5], which consists of a training set of nearly 5,500 questions, and a test set with 500 questions. The annotated question categories follow the question type taxonomy described in Section 2.1.

For all experiments regarding machine learning, we used Support Vector Machines (SVM), since it is largely stated in the literature as outperforming other machine learning techniques for question classification. Being so, we used the `LIBSVM` (Chang and Lin, 2001) implementation of a SVM classifier with the most appropriate kernels and optimized parameters, trained with all the 5,500 questions from the referred training set, using the one-versus-all multi-class strategy.

6.1 The performance of the Rule-based Classifier

The first experiment assessed the performance of the rule-based classifier and targeted the following answers:

- How many questions were (successfully and unsuccessfully) classified by the 60 manually built patterns described in Section 3 (from now on the *Patterns* method)?
- From the remaining questions, how many questions were (successfully and unsuccessfully) classified by the classification process based on the headword extraction, described in Section 4, and the subsequent mapping into the question classification category by using WordNet, as described in Section 5 (from now on the *Headword+WordNet* method)?
- In how many questions was the headword successfully extracted?
- How many questions were not classified by either of the previous processes?

These results were calculated for the coarse-grained categories, and also for the fined-grained ones (see Section 2.1 for details about the used taxonomy). Overall results are synthesized in Table 5.

It should be mentioned that in all unclassified questions – 38 questions – the headword was detected, but the WordNet process did not manage to find a category (obviously, all the unclassified questions were not matched by the *Pattern* method). For instance, although the headword (*birthstone*) was successfully found in the sentence

[5] `http://l2r.cs.uiuc.edu/~cogcomp/Data/QA/QC/`(last accessed in March 2010).

Results for coarse-grained categories								
Patterns			**Headword+WordNet**			**Overall Results**		
Right	Wrong	Prec.	Right	Wrong	Prec.	Prec.	Recall	Acc.
270	1	99.9%	165	26	86.4%	94.2%	92.4%	87.0%
Results for fine-grained categories								
Patterns			**Headword+WordNet**			**Overall Results**		
Right	Wrong	Prec.	Right	Wrong	Prec.	Prec.	Recall	Acc.
266	5	98.1%	150	41	78.5%	90.0%	92.4%	83.2%

Table 5 Rule-based classifier results for coarse- and fine-grained categories.

What is the birthstone for June?, no category was returned (because the word *birthstone* was not found in the WordNet).

Considering the set of the 26 question where the *Headword+WordNet* fails in the coarse-grained categorization, the headword was successfully extracted in 20 of these questions (meaning that the WordNet mapping was responsible for 20 failures in question classification).

Taking into account the previous 165 questions where the headword is successfully extracted, we can conclude that the headword extraction accuracy is 96.9%, which is a little higher that the results reported in (Metzler and Croft, 2005) (around 90%). Nevertheless, this is not a fair comparison as results were not obtained over the same corpora.

In a deeper analysis of the causes of the headword failure, it failed in description questions and also due to coordination. For instance, in the question *What is the Milky Way?*, *Way* was returned as the headword and in the question *What is bangers and mash?*, *mash* was the extracted headword.

We should also detach that the mapping through WordNet failed in 6 questions of type *What is the population...*. The word *population* is extracted as the headword, but by breadth-first search over the headword hypernyms lead to the classification Human and not Numeric, which was the correct one. By adding a simple rule to the manually built patterns would allow the *Patterns* method to solve this problem.

6.2 The Contribution of Unigrams

In this experiment we trained a SVM having only the question unigrams as features (from now on, we call U to this experiment). We choose unigrams, firstly due to its simplicity; secondly, because interesting results are described in the literature by using this feature (for instance (Huang et al., 2008) shows that unigrams help more in question classification that bigrams or trigrams). Results are the following:

- For coarse-grained categories, U accuracy is 88.8%;
- For fine-grained categories, U accuracy is 80.6%.

In a deeper analysis of results concerning the coarse-grained categories (similar conclusions can be taken for the fine grained categories), we observed the following:

- U has a better overall performance that the rule-base classifier for the coarse-grained categories;

- *U* fails in the same question that the *Patterns* method and it additionally fails in other 4 questions that the *Patterns* method is able to correctly classify;
- *U* fails in 36 questions that the *Headword+WordNet* is able to correctly classify, but *U* manages to correctly classify 6 questions that the *Headword+WordNet* is not able to correctly classify (although it tries);
- *U* correctly classifies 29 from the 38 questions that the rule-based classifier was enable to classify.

Thus, until this moment, the best results for coarse-grained categories would be obtained by using the rule-based question classifier to classify all the questions that it manages to classify, and then to use the SVM trained with unigrams to classify the remaining questions. That is, the rule-based classifier is more precise than *U*. By following this procedure, we would obtain an accuracy of 93%, as 270 question are correctly classified by *Patterns*, 165 questions are correctly classified by *HeadWord+WordNet* and 29 questions are correctly classified by *U*.

6.3 From Symbolic to Sub-symbolic Information

In a third experiment the information provided by the rule-based classifier – both headwords (**H**) and categories (**C**) – is used to generate the feature set for training, and merged with the information provided by the question unigrams (**U**). The obtained results are presented in Table 6.

Category Granularity	**H**	**C**	**H+C**	**U+H**	**U+C**	**U+H+C**
Coarse	63.2%	92.0%	94.6%	91.8%	**95.0%**	**95.0**%
Fine	39.0%	83.4%	88.8%	84.2%	90.2%	**90.8**%

Table 6 Accuracy of the machine-learning based classifier for coarse- and fine-grained categories.

The **H** feature holds the worst results, largely inferior to the ones achieved merely with the rule-based classifier. However, one should remember that only a small set of questions contains, in fact, a headword. In what concerns the machine learning classifier, when it is trained uniquely with feature **C**, results are similar to those obtained with the rule-based classifier (slightly better for the coarse-grained categories). It is when these features are combined with unigrams that the classifier holds the best results: an increase of 8.0% and 7.6% compared with the rule-based classifier, for coarse- and fine-grained categories, respectively. This lead us to conclude that the rule-based classifier was potentiated in a machine learning environment. It should be noticed that even without unigrams, features **H+C** already present an increase of accuracy from 87.0% to 94.6% for coarse-grained categories and from 83.2% to 88.8% for fine-grained categories, which is a particularly interesting result. In fact, this result is due to the following:

- The SVM learns Entity as a default value;

– The SVM is able to take advantage of the headword to find the correct classification in situations where the WordNet mapping failed. For instance, it is able to correctly classify as Location all the questions of type *Where is the capital of....*

6.4 Comparison With Other Works

We now compare our results with others reported in the literature for the question classification task. Table 7 summarizes the question classification accuracy reached by other relevant works, evaluated on the UIUC dataset.

	Category granularity	
Author	Coarse	Fine
This work	**95.0%**	**90.8%**
(Li and Roth, 2002)	91.0%	84.2%
(Zhang and Lee, 2003)	90.0%	80.2%
(Krishnan et al., 2005)	93.4%	86.2%
(Blunsom et al., 2006)	91.8%	86.6%
(Pan et al., 2008)	**94.0%**	-
(Huang et al., 2008)	**93.6%**	89.2%

Table 7 Comparison of accuracy results attained by this work, against other results reported in the literature that use the same dataset for training and testing.

Results show that by training a machine learning classifier with unigrams and the information manipulated by our rule-based classifier, achieves better accuracy for coarse- and fine-grained categories than the ones mentioned in state of the art literature.

Regarding coarse-grained classification, the work of (Pan et al., 2008) reported a similar result to ours, with an accuracy of 94%. As previously mentioned, the most striking feature from their work are the semantic classes, which can be seen as a simpler version of our headword feature.

With respect to fine-grained classification, the work of (Huang et al., 2008) obtains similar results to ours. These can be easily explained by the fact that we follow these authors as we also use headwords and their (semantic) classification as features.

Furthermore, it is also worth mentioning that our results are achieved using a compact feature space comprising nearly 10,000 distinct features, as against the 200,000 of (Li and Roth, 2002) and the 13,697 of (Huang et al., 2008), which results in a very efficient, yet effective question classifier.

7 Conclusions and Future Work

We presented and evaluated a rule-based question classifier, that follows two different strategies: either it performs a direct match into the question classification or it identifies the question headword and uses WordNet to map it into the question category. A set of 60 manually built rules were compiled for the direct match process; the headword is extracted according with another set of manually build rules that indicate the path in a parse tree towards the question's headword. The headword hypernyms are then analyzed, in a breadth-first search until one matches a set of names that is directly mapped into the possible question categories. Rules for the direct match process have a precision of 99.96% for coarse- and 98.1% for fine-grained questions. The process that involves headword extraction and WordNet mapping has a precision of 86.4% and 78.5% for coarse- and fine-grained questions respectively. Headword extraction has an accuracy of 96.9% for coarse-grained questions. Overall the rule-based classifier has a an accuracy of 87.0% for coarse- and 82.3% for fine-grained categories. To our knowledge it is the first rule-based question classifier that targets the widely used Li and Roth taxonomy. All the rules, algorithms and heuristics used in this process were detailed in the paper and all the material necessary to repeat the obtained results can be downloaded and used for research purposes.

The rule-based classifier – a symbolic system – was also used as a features' provider for a machine learning classifier – a sub-symbolic system –, being both its results and the information it manipulates used as features. Several experiments regarding the fusion of both symbolic and sub-symbolic systems are presented, and state of the art results in question classification are obtained when the information manipulated by the rule-based classifier is merged with unigrams, in the machine learning feature space. This process also resulted in an improvement of about 8% over the stand-alone rule-based classifier, which lead us to conclude that the performance of a rule-based question classifier can be enhanced simply by using it as a features' provider. How many rule-based classifiers can be improved in this way is a question that still needs to be answered, but the experiment is certainly worth to try.

As future work, we intend to use other types of semantic information such as semantic role labeling. Moreover, we will use the headwords extracted by the rule-based classifier in a query expansion process within a question answering system and we will use the cluster that aggregate similar synsets together to perform answer validation. This is certainly one of the advantages of having implemented a symbolic system: the possibility of re-using the involved information.

Acknowledgements Ana Cristina Mendes is supported by a PhD fellowship from Fundação para a Ciência e a Tecnologia (SFRH/BD/43487/2008).

References

Carlos Amaral, Adán Cassan, Helena Figueira, André Martins, Afonso Mendes, Pedro Mendes, Cláudia Pinto, and Daniel Vidal. Priberam's question answering system in QA@CLEF 2007. In *Advances in Multilingual and Multimodal Information Retrieval: 8th Workshop of the Cross-Language Evaluation Forum, CLEF 2007, Budapest, Hungary, September 19-21, 2007, Revised Selected Papers*, pages 364–371, Berlin, Heidelberg, 2008. Springer-Verlag. ISBN 978-3-540-85759-4. doi: http://dx.doi.org/10.1007/978-3-540-85760-0_46.

Rahul Bhagat, A. Leuski, and Eduard Hovy. Shallow semantic parsing despite little training data. Proceedings of the ACL/SIGPARSE 9th International Workshop on Parsing Technologies. Vancouver, B.C., Canada, 2005.

Phil Blunsom, Krystle Kocik, and James R. Curran. Question classification with log-linear models. In *SIGIR '06: Proceedings of the 29th annual international ACM*

SIGIR conference on Research and development in information retrieval, pages 615–616, New York, NY, USA, 2006. ACM. ISBN 1-59593-369-7. doi: http://doi.acm.org/10.1145/1148170.1148282.

Andrew J. Carlson, Chad M. Cumby, Jeff L. Rosen, and Dan Roth. Snow user guide. Technical Report UIUC-DCS-R-99-210, Champaign, IL, USA, 1999.

Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Michael John Collins. *Head-driven statistical models for natural language parsing*. PhD thesis, Philadelphia, PA, USA, 1999.

C. Fellbaum, editor. *WordNet: An Electronic Lexical Database*. MIT Press, 1998. URL `http://books.google.es/books?hl=es&lr=&id=Rehu8OOzMIMC&oi=fnd&pg=PR11`.

Ulf Hermjakob, Eduard Hovy, and Chin-Yew Lin. Automated question answering in Webclopedia: a demonstration. In *Proceedings of the second international conference on Human Language Technology Research*, pages 370–371, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.

Zhiheng Huang, Marcus Thint, and Zengchang Qin. Question classification using head words and their hypernyms. In *EMNLP*, pages 927–936, 2008.

John Judge, Aoife Cahill, and Josef van Genabith. Questionbank: creating a corpus of parse-annotated questions. In *ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 497–504, Morristown, NJ, USA, 2006. Association for Computational Linguistics. doi: http://dx.doi.org/10.3115/1220175.1220238.

Vijay Krishnan, Sujatha Das, and Soumen Chakrabarti. Enhanced answer type inference from questions using sequential models. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 315–322, Morristown, NJ, USA, 2005. Association for Computational Linguistics. doi: http://dx.doi.org/10.3115/1220575.1220615.

Cody C. T. Kwok, Oren Etzioni, and Daniel S. Weld. Scaling question answering to the web. In *WWW '01: Proceedings of the 10th international conference on World Wide Web*, pages 150–161, New York, NY, USA, 2001. ACM. ISBN 1-58113-348-0. doi: http://doi.acm.org/10.1145/371920.371973.

Fangtao Li, Xian Zhang, Jinhui Yuan, and Xiaoyan Zhu. Classifying what-type questions by head noun tagging. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 481–488, Manchester, UK, August 2008. Coling 2008 Organizing Committee. URL `http://www.aclweb.org/anthology/C08-1061`.

Xin Li and Dan Roth. Learning question classifiers. In *Proceedings of the 19th international conference on Computational linguistics*, pages 1–7, Morristown, NJ, USA, 2002. Association for Computational Linguistics. doi: http://dx.doi.org/10.3115/1072228.1072378.

Ana Mendes, Lusa Coheur, Nuno J. Mamede, Ricardo Daniel Ribeiro, David Martins de Matos, and Fernando Batista. QA@L2F, first steps at QA@CLEF. In *Advances in Multilingual and Multimodal Information Retrieval*, volume 5152 of *Lecture Notes in Computer Science*. Springer-Verlag, September 2008.

Donald Metzler and W. Bruce Croft. Analysis of statistical question classification for fact-based questions. *Inf. Retr.*, 8(3):481–504, 2005.

Dan Moldovan, Marius Paca, Sanda Harabagiu, and Mihai Surdeanu. Performance issues and error analysis in an open-domain question answering system. *ACM Trans.*

Inf. Syst., 21(2):133–154, 2003.

Dan I. Moldovan, Sanda M. Harabagiu, Marius Pasca, Rada Mihalcea, Roxana Girju, Richard Goodrum, and Vasile Rus. The structure and performance of an open-domain question answering system. In *ACL*, 2000.

Allen Newell. *Unified Theories of Cognition.* Harvard University Press, 1990.

Yan Pan, Yong Tang, Luxin Lin, and Yemin Luo. Question classification with semantic tree kernel. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 837–838, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-164-4. doi: http://doi.acm.org/10.1145/1390334.1390530.

Slav Petrov and Dan Klein. Improved inference for unlexicalized parsing. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 404–411, Rochester, New York, April 2007. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/N/N07/N07-1051`.

E. Saquete, J. L. Vicedo, P Martínez-Barco, R. Munoz, and H. Llorens. Enhancing qa systems with complex temporal question processing capabilities. *Journal of Artificial Intelligence Research*, 35:299–330, 2009.

B. A. Sharada and P. M. Girish. Wordnet has no 'recycle bin', 2004.

Herbert A. Simon. *The Sciences of the Artificial.* MIT Press, Cambridge, Massachusetts, first edition, 1969.

Ron Sun. A two-level hybrid architecture for structuring knowledge for commonsense reasoning. In Ron Sun and Lawrence A. Bookman, editors, *Computational Architectures Integrating Neural and Symbolic Processing*, chapter 8, pages 247–182. Kluwer Academic Publishers, 1995.

Alfred Tarski. *Logic, Semantics,Metamathematics.* Oxford University Press, London, 1956.

Dell Zhang and Wee Sun Lee. Question classification using support vector machines. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 26–32, New York, NY, USA, 2003. ACM. ISBN 1-58113-646-3. doi: http://doi.acm.org/10.1145/860435.860443.