Dialogue and Discourse 3(2) (2012) 147-175 doi: 0.5087/dad.2012.207

Question Generation based on Lexico-Syntactic Patterns Learned from the Web

Sérgio Curto SERGIO.CURTO@L2F.INESC-ID.PT
Spoken Language Systems Laboratory - L²F/INESC-ID
R. Alves Redol, 9 - 2º– 1000-029 Lisboa, Portugal

Ana Cristina Mendes ANA.MENDES@L2F.INESC-ID.PT
Spoken Language Systems Laboratory - L²F/INESC-ID
Instituto Superior Técnico, Technical University of Lisbon
R. Alves Redol, 9 - 2º– 1000-029 Lisboa, Portugal

Luísa Coheur LUISA.COHEUR@L2F.INESC-ID.PT
Spoken Language Systems Laboratory - L²F/INESC-ID
Instituto Superior Técnico, Technical University of Lisbon
R. Alves Redol, 9 - 2º– 1000-029 Lisboa, Portugal

Editor: Paul Piwek and Kristy Elizabeth Boyer

Abstract

THE-MENTOR automatically generates multiple-choice tests from a given text. This tool aims at supporting the dialogue system of the FalaComigo project, as one of FalaComigo's goals is the interaction with tourists through questions/answers and quizzes about their visit. In a minimally supervised learning process and by leveraging the redundancy and linguistic variability of the Web, THE-MENTOR learns lexico-syntactic patterns using a set of question/answer seeds. Afterwards, these patterns are used to match the sentences from which new questions (and answers) can be generated. Finally, several filters are applied in order to discard low quality items. In this paper we detail the question generation task as performed by THE-MENTOR and evaluate its performance.

Keywords: Question Generation, Pattern Learning, Pattern Matching

1. Introduction

Nowadays, interactive virtual agents are a reality in several museums worldwide (Leuski et al. 2006, Kopp et al. 2005, Bernsen and Dybkjær 2004). These agents have educational and entertainment goals – "edutainment" goals, as defined by Adams et al. (1996) – and communicate through natural language. They are capable of establishing small social dialogues, teaching some topics for which they were trained, and asking/answering questions about these topics. Following this idea, the FalaComigo project invites tourists to interact with virtual agents through questions/answers and multiple-choice tests.[1] Agent Edgar (Figure 1) is the face of the project, currently operating in the Palace of Monserrate (Moreira et al. 2011).

1. A multiple-choice test is composed of several multiple-choice test items, each item consisting of a question, its correct answer and a group of incorrect answers (also called distractors).

 Submitted 2/11; Accepted 1/12; Published online 3/12

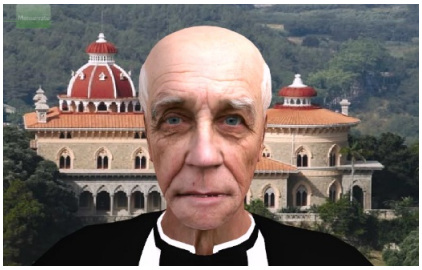


Figure 1: Agent Edgar

The efforts of enhancing the Portuguese cultural tourism started with DuARTE Digital (Mendes et al. 2009), a virtual agent based on the DIGA dialog framework (Martins et al. 2008), which engages in inquiry-oriented conversations with users, answering questions about a famous piece of Portuguese jewelry, Custódia de Belém. The approach based on multiple-choice tests has also been recently explored in the Lisbon Pantheon, where a virtual agent assesses the visitors' knowledge about the monument. However, in all these cases, the involved questions/answers and multiple-choice tests are hand crafted, which poses a problem every time a different monument or piece of art becomes the focus of the project or a different language is to be used. Therefore, one of Fala-Comigo's main objectives is to find a solution to automatically generate questions, answers and distractors from texts describing the monument or piece of art in study, without needing to have experts hand coding specific rules for this purpose. THE-MENTOR (Mendes et al. 2011) is our response to this challenge, as it receives as input a set of seeds – that is, natural language question/answer pairs – and uses them to automatically find, in the Web, lexico-syntactic patterns capable of generating new question/answer pairs, as well as distractors from a given a text (a sentence, a paragraph or an entire document). Contrary to other systems, the process of building these patterns is automatic, based on the previous mentioned set of seeds. Thus, as done in many Information Extraction frameworks – such as DARE (Domain Adaptive Relation Extraction)[2](Xu et al. 2010) – our approach is based on minimally supervised learning, where a set of seeds is used to bootstrap patterns that will be used to find the target type of information.

Figure 2 illustrates some of the information used by THE-MENTOR. The first line shows the syntactic structure of a seed question; the second line depicts one learned lexico-syntactic pattern. Different colors are used to identify the matching between the generated pattern and the text tokens. The last line shows an example of a multiple-choice test item created by THE-MENTOR.[3]

In this paper, we present our approach to the text-to-question task (as identified by Rus et al. (2010a)) and we thoroughly describe and evaluate the factoid question generation as performed by THE-MENTOR.

The remainder of this paper is organized as follows: in Section 2 we overview related work. In Sections 3 and 4 we detail the pattern learning and question generation subtasks, respectively. In Section 5 we evaluate THE-MENTOR. Finally, in Section 6 we present the main conclusions and point out future work directions.

2. http://dare.dfki.de/.

3. A first version of THE-MENTOR is online at http://services.l2f.inesc-id.pt/the-mentor. In http://qa.l2f.inesc-id.pt/ one can find the corpora and scripts used/developed for the evaluation of THE-MENTOR. The code of THE-MENTOR will be made available in the near future.

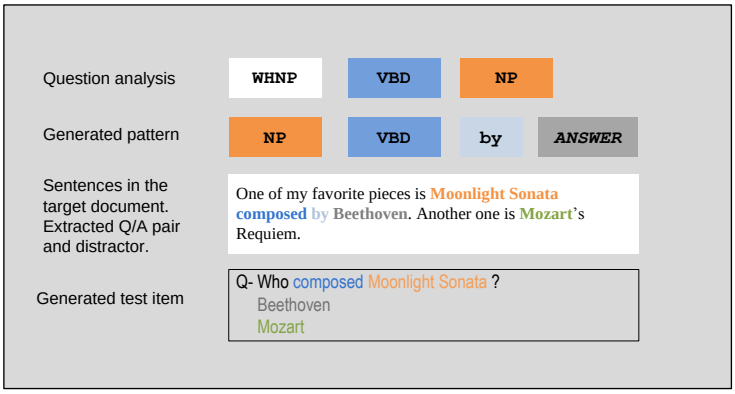


Figure 2: Information manipulated by THE-MENTOR.

2. Related Work

Question Generation (QG) is, nowadays, an appealing field of research. Three recent workshops exclusively dedicated to QG and a first shared evaluation challenge, in 2010, with the goal of generating questions from paragraphs and sentences (Rus et al. 2010a) have, definitively, contributed to the increase of interest in this topic.

Areas such as Question-Answering (QA), Natural Language Generation or Dialogue Systems contribute to QG with techniques and resources, such as corpora with examples of questions and grammars designed to parse/generate questions. Nevertheless, the specificities of QG raise many issues, namely the taxonomy that should be considered, the process of generation or the systems' evaluation. These and many others questions led to the actual research lines. In the following we present some recent discussions and achievements related to the different research targets in QG.

One of the decisions that needs to be made has to do with choosing the question taxonomy that will be used during the QG. As said by Forăscu and Drăghici (2009), in the context of a QG campaign, it is important to have a clear inventory of the most used question types. Forăscu and Drăghici (2009) survey several taxonomies proposed in the literature and point to some available resources. Boyer et al. (2009) propose a hierarchical classification scheme for tutorial questions, in which the top level identifies the tutorial goal and the second level the question type, sharing many categories with previous work on classification schemes. Despite the differences between this and other taxonomies existing in the literature, it should be clear that the decision of choosing a taxonomy is directly related to the goal of the task in hands.

Another aspect that needs to be carefully addressed in QG is related to the information sources in use. In fact, if it is obvious that the decision of using a certain resource should be influenced by the specific goal of the QG system (for instance, if the target are questions about politics, the information source should be of that domain), it is also true that different texts in the same domain can involve different difficulty levels of processing and, thus, lead to better or worse questions. For instance, a text about art in Wikipedia is probably easier to process than a text about the same subject written by experts for experts. Since Wikipedia articles are to be read by everybody, they usually have an accessible vocabulary and simple syntax; a text written by experts for experts tends to use specific vocabulary and they are not written with the goal of being understood by everyone. Therefore, a rule-based QG system will probably have more difficulty when generating plausible questions from

the latter. Therefore, in some QG systems (and as done in many other Natural Language Processing tasks), texts are simplified in a pre-processing step, before being used for QG. For instance, the approach of Heilman and Smith (2010) – where an algorithm for extracting simplified sentences from a set of syntactic constructions is presented – texts are simplified before being submitted to the QG module. Authors show that this process results in a more suitable generation of factoid questions. Kalady et al. (2010) also report a text pre-processing step before the question generation stage, where anaphora resolution and pronoun replacement is performed.

Still related to the type of text to be used as input, a change in the nature of the information source can lead to extensive changes in the QG system, as reported by Chen et al. (2009), the authors of Project LISTEN's Reading Tutor, where an intelligent tutor tries to scaffold the self-questioning strategy that readers usually use when reading a text. Chen et al. (2009) explain the changes they had to do in their approach when they moved from narrative fiction to information texts.

Concerning resources, many systems perform QG over Wikipedia (Heilman and Smith 2009). However, Becker et al. (2009) execute their QG system over the Full Option Science System (FOSS) and Wyse and Piwek (2009) draw the attention of the community to the OpenLearn repository, which covers a wide range of materials (in different formats) and is only authored by experts. In the latter, a QG system is implemented, Ceist, running on the OpenLearn repository. Corpora from this repository were also used in the evaluation of the First Question Generation Shared Task and Evaluation Challenge (QGSTEC 2010). In fact, another important resource for the QG community are the corpora provided by the CODA project (Piwek and Stoyanchen 2010) where, in addition to a development/test set with questions and answers, a dialogue corpus is provided, where segments are aligned with monologue snippets (with the same content). The CODA project team targets a system for automatically converting monologue into dialogue. In their approach, a set of rules maps discourse relations marked in the (monologue) corpus to a sequence of dialogue acts. Also, as suggested by Ignatova et al. (2008) sites such as Yahoo! Answers[4] or WikiAnswers[5] can have an important role in QG, namely in tasks such as generating high quality questions from low quality ones.

Regarding the QG process, different approaches have been followed. The usage of patterns to bridge the gap between the question and the sentence in which the answer can be found is a technique that has been extensively used in QA and in QG. The main idea behind this process is that the answer to a given question will probably occur in sentences that contain a rewrite of the original question. In the QA track of the TREC-10, the winning system – described in (Soubbotin 2001) – presents an extensive list of surface patterns. The majority of systems that target QG also follow this line of work and are based on handmade rules that rely on pattern matching to generate questions. For instance, in (Chen et al. 2009), after the identification of key points, a situation model is built and question templates are used to generate questions from those situation models. The Ceist system described in (Wyse and Piwek 2009) uses syntactic patterns and a tool called Tregex (Levy and Andrew 2006) that receives as input a set of hand-crafted rules and matches these rules against the parsed text, generating, in this way, questions (and answers). Kalady et al. (2010) bases their QG process in Up-keys – that are significant phrases in documents – in parse tree manipulation and Named Entity Recognition (NER). Up-keys are used to generate definitional questions; Tregex is, again, used for parse tree manipulation (as well as Tsurgeon (Levy and Andrew 2006)); named entities are used to generate factoid questions, according to their semantics. For instance, if a named

4. http://answers.yahoo.com/
5. http://wiki.answers.com/

entity of type PEOPLE is detected, a Who-question is triggered (if some conditions are verified). Heilman and Smith (2009) also consider a transformation process in order to generate questions.

Considering the evaluation of QG systems, several methods have been proposed. Chen et al. (2009), for instance, classify the generated questions as plausible or implausible, according to their grammatical correctness and if they make sense in the context of the text. Becker et al. (2009) give a detailed evaluation of their work, where automatically generated questions are compared against questions generated by humans (experts and not experts). They concluded that questions from human tutors outscore their system. The computer-aided environment for generating multiple-choice test items, described in Mitkov et al. (2006) also proposes a similar evaluation, where multiple-choice tests generated by humans are compared with the ones generated by their system. The authors start by identifying and extracting key-terms from the source corpora, using regular expressions that match nouns and noun-phrases; afterwards, question generation rules are applied to sentences with specific structures; finally, a filter assures the grammatical correctness of the questions, although in a post-editing phase, results are revised by human assessors. This system was later adapted to the medical domain (Karamanis et al. 2006). The work described by Heilman and Smith (2009) should also be mentioned, as they propose a method to rank generated questions, instead of classifying them.

3. Learning Patterns

In THE-MENTOR, a set of question/answer pairs is used to create the seeds that will be the input of a pattern learning process. Its pattern learning algorithm is similar to the one described by Ravichandran and Hovy (2002). For a given question category, their algorithm learns lexical patterns (instead, our patterns also contain syntactic information) between two entities: a question term and the answer term. These entities are submitted to Altavista and, from the 1000 top retrieved documents, the ones containing both terms are kept. The patterns are the longest matching substrings extracted from these documents, given that the question term and the answer term are replaced by the tokens `<NAME>` and `<ANSWER>`, respectively. For instance, for the category BIRTHDATE, a learned pattern is `<NAME> was born in <ANSWER>`. Some examples of this line of work are the QA systems described by Brill et al. (2002) and Figueroa and Neumann (2008). The former bases the performance of the QA system AskMSR on manually created rewrite rules which are likely substrings of declarative answers to questions; the latter searches for possible answers by analysing substrings that have similar contexts of already known answers and uses genetic algorithms in the process. Brill et al. (2002) also explain the need to produce less precise rewrites, since the correct ones did not match any document. We also feel this need and THE-MENTOR also accepts patterns that do not match all the components of the question.

In the following section we describe the component responsible for learning lexico-syntactic patterns in THE-MENTOR. We start by explaining how THE-MENTOR builds seed/validation pairs that originate the patterns. Afterwards, we present the actual process of pattern learning and describe the three types of learned patterns: `strong`, `inflected` and `weak`. Finally, we end with a description of how the learned patterns are validated.

3.1 Building seed/validation pairs

THE-MENTOR starts by building a group of seed/validation pairs based on a set of questions and their respective correct answers (henceforward, a pair composed of a question and its answer will

be denoted as Question/Answer (Q/A) pair). Figure 3 shows examples of Q/A pairs given as input to THE-MENTOR.

"*In which city is the Wailing Wall?*; *Jerusalem*"
"*In which city is Wembley Stadium?*; *London*"
"*In which sport is the Cy Young Trophy awarded?*; *baseball*"
"*In which sport is the Davis Cup awarded?*; *tennis*"
"*Who painted the Birth of Venus?*; *Botticelli*"
"*Who sculpted the Statue of David?*; *Michelangelo*"
"*Who invented penicillin?*; *Alexander Fleming*"
"*Where was the first Zoo?*; *China*"
"*Where was the Liberty Bell made?*; *England*"
"*How many years did Rip Van Winkle sleep?*; *twenty*"
"*How many years did Sleeping Beauty sleep?*; *100*"

Figure 3: Questions and their correct answers used as input to THE-MENTOR.

A seed/validation pair is composed by two Q/A pairs: the first pair is used to query the selected search engine and extract patterns from the retrieved results – the seed Q/A pair; the second pair validates the learned patterns by instantiating them with the content of its question – the validation Q/A pair. For the pattern learning subtask, the questions in a seed/validation pair are required to have the same syntactic structure and prefix, and the answers are supposed to be of the same category. For instance, ["*How many years did Rip Van Winkle sleep?*; *twenty*"/"*How many years did Sleeping Beauty sleep?*; *100*"] is a seed, while ["*Who pulled the thorn from the Lion's paw?*; *androcles*"/"*What was the color of Christ's hair in St John's vision?*; *white*"] is not.

In order to create seed/validation pairs from the given set of Q/A pairs, THE-MENTOR performs three steps:

1. All questions are syntactically analyzed. This step allows to group Q/A pairs that are syntactically similar and avoiding the creation of seed composed by pairs like "*How many years did Rip Van Winkle sleep?*; *twenty*" and "*Who sculpted the Statue of David?*; *Michelangelo*" (see Table 1). In this work, we used the Berkeley Parser (Petrov and Klein 2007) trained on the QuestionBank (Judge et al. 2006), a treebank of 4,000 parse-annotated questions;

Index	0	1	2	3
Tag	WHNP	VBD	NP	VB
Content	*how many years*	*did*	*rip van winkle*	*sleep*
Index	0	1	2	3
Tag	WHNP	VBD	NP	
Content	*who*	*sculpted*	*the Statue of David*	

Table 1: Syntactic constituents identified by the parser for two input questions.

2. The content of the first syntactic constituent of the question (the question's prefix) is collected. This is a first step towards a semantically-driven grouping of Q/A pairs, allowing the creation of pairs like ["*What color was the Maltese Falcon?*; *black*"/"*What color was Moby Dick?*;

white"]. However, it also leads to wrongly grouped pairs like ["*What was the language of Nineteen Eighty Four?*; *newspeak*"/"*What was the color of Christ's hair in St John's vision?*; *white*"] (the question's prefixes are *what color* and *what*, respectively);

3. All questions are classified according to the category of the expected answer. This step allows us to group the Q/A pairs when the answers share the same semantic category, such as "*Who painted the Birth of Venus?*; *Botticelli*" and "*Who sculpted the Statue of David?*; *Michelangelo*", both expecting a person's name as answer. In this work, a machine learning-based classifier was used, and fed with features derived from a rule-based classifier (Silva et al. 2011). The used taxonomy is Li and Roth's two-layer taxonomy (Li and Roth 2002), consisting of a set of six coarse-grained categories (ABBREVIATION, ENTITY, DESCRIPTION, HUMAN, LOCATION and NUMERIC) and fifty fine-grained ones. This taxonomy is widely used by the machine learning community (Silva et al. 2011, Li and Roth 2002, Blunsom et al. 2006, Huang et al. 2008, Zhang and Lee 2003), because the authors have published a set of nearly 6,000 labeled questions (the University of Illinois at Urbana-Champaign dataset), freely available on the Web, making it a very valuable resource for training and testing machine learning models.

After these steps, the Q/A pairs are divided into different groups according to their syntactic structure, prefix and category, and THE-MENTOR builds seed/validation pairs from all combinations of two Q/A pairs in every group. Thus, if a group has n Q/A pairs, THE-MENTOR builds $\binom{n}{2} = \frac{n!}{2(n-2)!}$ seed/validation pairs.

The following are examples of built seed/validation pairs, grouped according to the syntactic structure and category of their questions:

LOCATION:CITY –WHPP VBZ NP

["*In which city is the Wailing Wall?*; *Jerusalem*"/"*In which city is Wembley Stadium?*; *London*"]

HUMAN:INDIVIDUAL – WHNP VBD NP

["*Who painted the Birth of Venus?*; *Botticelli*"/"*Who sculpted the Statue of David?*; *Michelangelo*"]

["*Who sculpted the Statue of David?*; *Michelangelo*"/"*Who invented penicillin?*; *Alexander Fleming*"]

["*Who invented penicillin?*; *Alexander Fleming*"/"*Who painted the Birth of Venus?*; *Botticelli*"]

ENTITY:WORD – WHNP VBZ NP

["*What is the last word of the Bible?*; *amen*"/"*What is the Hebrew word for peace used as both a greeting and a farewell?*; *shalom*"]

HUMAN:INDIVIDUAL – WHNP VBD NP

["*Who was Don Quixote's imaginary love?*; *dulcinea*"/"*Who was Shakespeare's fairy king?*; *oberon*"].

3.2 Query Formulation and Passage Retrieval

THE-MENTOR captures the frequent patterns that relate a question to its correct answer from the Web. The extraordinary dimensions of this particular document collection allow the existence of certain properties that are less expressive in other built document collections: the same information is likely to be replicated multiple times (redundancy) and presented in various ways (linguistic variability). THE-MENTOR explores these two properties – redundancy and linguistic variability – to learn patterns from a seed/validation pair.

The process of accessing the Web relies on submissions to a Web search engine of several queries. For a given seed Q/A pair, different queries are built from the permutations of the set composed by: (a) the content of the phrase nodes (except the *Wh*-phrase) of the question, (b) the answer and (c) a wildcard (*).

THE-MENTOR uses phrase nodes instead of words (as done by Ravichandran and Hovy (2002)) since they allow us to reduce the number of permutations and, thus, of query submissions to the chosen search engine; also, they usually represent a single unit of meaning, and therefore should not be broken down into parts (except for verb phrases). For instance, considering the question *Who sculpted the Statue of David?*, the corresponding parse tree and phrasal nodes are depicted in Figure 4 (we added a numeric identifier to the tree nodes to help their identification.). It does not make sense to divide the noun phrase *the Statue of David*, as it would generate several meaningless permutations, such as `Statue the sculpted Michelangelo of * David`.

The wildcard is used as a placeholder for one or more words and allows diversity in the learned patterns.[6] For instance, the wildcard in the query `"Michelangelo * sculpted the statue of David"` allows the sentence *Michelangelo has sculpted the statue of David* to be present in the search results, matching the verb *has*.

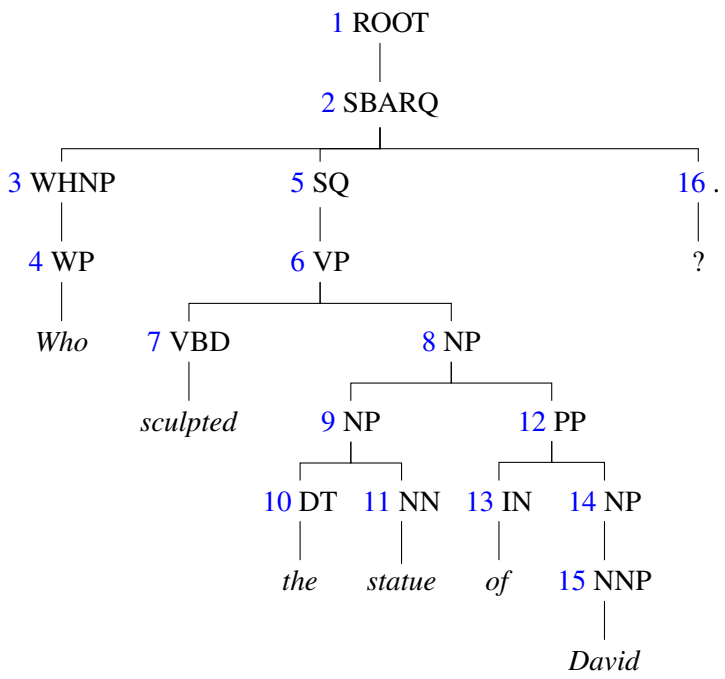
One query is built from one permutation enclosed in double quotes, which will constrain the search results to contain the words in the query in that exact same order, without any other change. For example, the query `"sculpted the Statue of David * Michelangelo"` requires the presence in the search results of the words exactly as they are stated in the query. Here the wildcard stands for one or more extra words allowed between the tokens *David* and *Michelangelo*. The total number of permutations (queries submitted) is $n! - 2(n-1)!$, in which n is the number of elements to be permuted.

Finally, the query is sent to a search engine.[7] Both `"Michelangelo * sculpted the statue of David"` and `"the statue of David sculpted * Michelangelo"` are examples of two queries submitted to the search engine for the seed Q/A pair "*Who sculpted the statue of David?*; *Michelangelo*".

The brief summaries, or passages, retrieved by the search engine to the submitted query are then used to learn patterns. Being so, the passages[8] are broken down into sentences, parsed (we have used, once again, the Berkeley Parser, this time trained with the Wall Street Journal) and if there is a text fragment that matches the respective permutation, we rewrite it as a pattern. The pattern is composed of the lexico-syntactic information present in the matched text fragment, thus, expressing the relation between the question and its answer present in the sentence.

6. The wildcard is not allowed as the first or the last element of the permutation and, thus, of the query.
7. In this work, we use Google. However, there is no technical reason that hinders THE-MENTOR from using other search engine.
8. For now, we ignore the fact that passages are often composed of incomplete sentences.

(a)



(b)

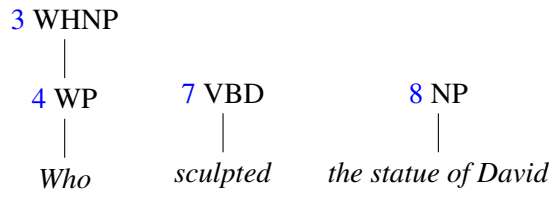


Figure 4: (a) Parse tree of the question *Who sculpted the Statue of David?*. (b) Phrase nodes of the parse tree in (a).

Algorithm 1 describes the pattern learning algorithm, which includes the procedures to build the query, the call to the search engine and the reformulation of text fragments into patterns.

Consider again the question *Who sculpted the Statue of David?*, its flat syntactic structure "[$_{\text{WHNP}}$ Who] [$_{\text{VBD}}$ sculpted] [$_{\text{N}}$ the Statue of David]", its correct answer *Michelangelo*, the sentence *Michelangelo has sculpted the statue of David* (that matches the permutation `"Michelangelo * sculpted the statue of David"`) and its parse tree depicted in Figure 5. A resulting pattern would be `NP`{*ANSWER*} `[has] VBD NP`, where: the tag {*ANSWER*} indicates the position of the answer; the syntactic information is encoded by the tags of the parsed text fragment; and, the lexical information is represented by the tokens between square brackets. Note that the lexical information that composes the patterns refers to the tokens learned from the sentence, but that are not present in the seed question, nor in the answer.

Algorithm 1 Pattern learning algorithm

```
procedure PATTERN-LEARNING(seed-pair : question-answer pair)
    patterns ← []
    phrase-nodes ← GET-PHRASE-NODES(seed-pair.question.parse-tree)
    for each permutation in PERMUTE({phrase-nodes, *, seed-pair.answer}) do
        query ← ENCLOSE-DOUBLE-QUOTES(permutation)
        results ← SEARCH(query)
        for each sentence in results.sentences do
            if MATCHES(sentence, permutation) then
                pattern ← REWRITE-AS-PATTERN(sentence, phrase-nodes)
                patterns ← ADD(patterns, pattern)
            end if
        end for
    end for
    return patterns
end procedure
```

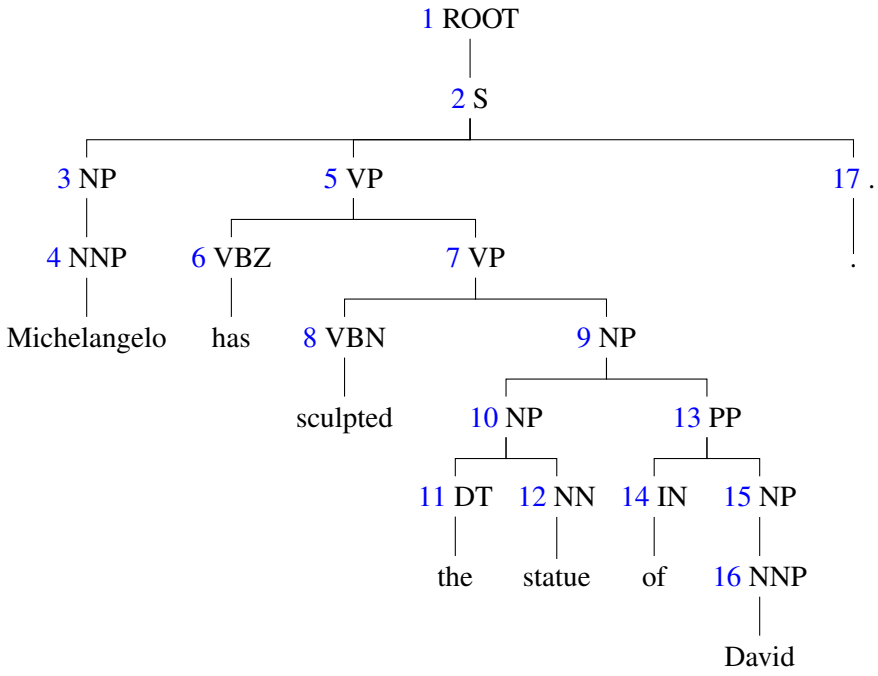


Figure 5: Parse tree of the sentence *Michelangelo has sculpted the statue of David.*

3.3 Strong, inflected and weak patterns

Patterns that withhold the content of all phrases of the seed question are named `strong` patterns. These are built by forcing every phrase (except the *Wh*-phrase) to be present in the patterns. This is the expected behaviour for noun- and prepositional-phrases that should be stated *ipsis verbis* in the sentences from where the patterns are learned. However, the same does not apply for verb-

phrases. For instance, the system should be flexible enough to learn the pattern for the question *Who sculpted the Statue of David?* from the sentence *Michelangelo finished sculpting the statue of David in 1504*, even though the verb *to sculpt* is inflected differently. Therefore, another type of patterns is allowed – the `inflected` patterns – where the main verb of the question is replaced by each of its inflections and the auxiliary verb (if it exists) is removed from the permutations that build the queries sent to the search engine. Being so, the next queries are also created and submitted to the search engine: "`Michelangelo * sculpting the statue of David`", "`Michelangelo * sculpt the statue of David`" and "`Michelangelo * sculpts the statue of David`". This allows the system to find patterns that are not in the same verbal tense of the question.

A final type of patterns – the `weak` patterns – is also possible. In order to learn these patterns, only the noun- and prepositional-phrases of the question, the answer and the wildcard are submitted to the search engine, that is, verb-phrases are simply discarded. Following the previous example, both "`Michelangelo * the statue of David`" and "`the statue of David * Michelangelo`" are the two queries that might result in `weak` patterns. Although they do not completely rephrase the question, these patterns are particularly interesting because they can capture the relation between the question and the answer. For instance, the pattern `NP [,] [by] NP{`*`ANSWER`*`}` should be learned from the sentence *The Statue of David, by Michelangelo*, even if it does not include the verb (in this case, *sculpted*). These patterns are different from the `strong` and `inflected` patterns, not only because of the way they were created, but also because they trigger distinct strategies to question generation, as we will see.

Finally, we should mention that THE-MENTOR patterns are more complex than the ones presented, as they are explicitly linked to their seed question by indexes, mapping the position of each one of their components into the seed question components. For the sake of simplicity, we decided to omit these indices.[9] The syntax of the patterns, however, is not as sophisticated as the ones allowed by Tregex (Levy and Andrew 2006), since it does not allow so many node-node relations.

3.4 Patterns' Validation

The first Q/A pair of each seed/validation pair – the seed Q/A pair – is necessary for learning `strong`, `inflected` and `weak` patterns, as explained before. However, these patterns have to be validated by the second Q/A pair. This validation is required because, although many of the learned patterns are generic enough to be applied to other questions, others are too closely related to the seed pair and, therefore, too specific. For instance, the pattern `NP [:] NNP{`*`ANSWER`*`} ['s] [Renaissance] [masterpiece] [was] VBN` (learned from the sentence *Statue of David: Michelangelo's Renaissance masterpiece was sculpted from 1501 to 1504.*) is specific to a certain cultural movement (the Renaissance) and will only be applied in sentences that refer to it. Therefore, the validation Q/A pair of the seed/validation pair is parsed, matched against the previously learned patterns and sent to the search engine.

For a candidate pattern to be validated, its components are instantiated with the content of the question's respective syntactic constituents. The result of this instantiation is enclosed in double quotes and sent to the search engine. A score that measures the generality of the pattern is given by the ratio between the number of retrieved text snippets and the maximum number of text snippets retrieved by the search engine. If this score is above a certain threshold, the pattern is validated.

9. We will refer to these indices in Section 4, which is dedicated to the question generation subtask.

For example, consider the seed/validation pair ["*Who sculpted the Statue of David?*; *Michelangelo*"/"*Who invented penicillin?*; *Alexander Fleming*"], that WHNP VBD NP is the parsing result of the question in the second Q/A pair and that `NP VBD by {ANSWER}` is a candidate pattern. The result of the instantiation of the pattern, later submitted to the search engine, is: `"penicillin invented by Alexander Fleming"`. This strategy is possible because THE-MENTOR forces both questions of a seed/validation pair to be syntactically similar (WHNP VBD NP is also the parsing result of the question in the first Q/A pair). Therefore, the components of the candidate pattern can be associated with the syntactic constituents of the second Q/A pair.

Algorithm 2 summarizes this step. In this work, the maximum number of text snippets was 16 and the threshold was set to 0.25.

Algorithm 2 Pattern validation algorithm

procedure PATTERN-VALIDATION($patterns$: patterns to validate, $validation\text{-}pair$: question-answer validation pair)
  $threshold \leftarrow X$
  $validated\text{-}patterns \leftarrow []$
  for each $pattern$ **in** $patterns$ **do**
    $query \leftarrow$ REWRITE-AS-QUERY($validation\text{-}pair, pattern$)
    $query \leftarrow$ ENCLOSE-DOUBLE-QUOTES($query$)
    $results \leftarrow$ SEARCH($query$)
    $score \leftarrow results.size/max\text{-}size$
    if $score \geq threshold$ **then**
      $validated\text{-}patterns \leftarrow$ ADD($validated\text{-}patterns, pattern$)
    end if
  end for
  return $validated\text{-}patterns$
end procedure

Table 2 shows a simplified example of a set of patterns (with different types) learned for questions with syntactic structure WHNP VBD NP and category HUMAN:INDIVIDUAL.

Question HUMAN:INDIVIDUAL-WHNP VBD NP

Score	Pattern	Type
0.625	{ANSWER} ['s] NP	Weak
0.9375	{ANSWER} [painted] NP	Weak
0.25	{ANSWER} [began] VBG NP	Inflected
1.0	{ANSWER} [to] VB NP	Inflected
0.625	NP VBD [by] {ANSWER}	Strong
1.0	{ANSWER} VBD NP	Strong

Table 2: Examples of learned patterns with respective score and type.

4. Question Generation

In this section we describe the component responsible for generating questions in THE-MENTOR. We start by presenting our algorithm for matching the lexico-syntactic patterns. Afterwards, we show how question generation is performed depending on the type of learned patterns and we finish with a description of the filters used to discard generated questions of low quality.

4.1 Pattern matching

The subtask dedicated to the question (and answer) generation takes as input the previously learned set of lexico-syntactic patterns and a parsed target text (a sentence, paragraph or a full document) from which questions should be generated. Afterwards, the patterns are matched against the syntactic structure of the target text.

For a match between a fragment of the target text and a pattern to occur, there must be a lexico-syntactic overlap between the pattern and the fragment. Thus, two conditions must be met:

- the fragment of the target text has to share the syntactic structure of the pattern. That is, in a depth-first search of the parse tree of the text, the sequence of syntactic tags in the pattern has to occur; and
- the fragment of the target text has to contain the same words of the pattern (if they exist) in that exact same order.

This means that each match is done both at the lexical level – since most of the patterns include surface words – and at the syntactic level. For that purpose, we have implemented a (recursive) algorithm that explores the parsed tree of a text in a top-down, left-to-right, depth-first search. This algorithm tests if the components of the lexico-syntactic pattern are present in any fragment of the target text.

For example, consider the `inflected` pattern `NP{`*`ANSWER`*`} [began] VBG NP`, a sentence *In November 1912, Kafka began writing The Metamorphosis* in the target text and its parse tree (Figure 6).

In this case, there are two text fragments in the sentence that syntactically match the learned pattern, namely those conveyed by the following phrase nodes:

- 5 NP 12 VBD 15 VBG 16 NP – *November 1912 began writing The Metamorphosis*
- 9 NP 12 VBD 15 VBG 16 NP – *Kafka began writing The Metamorphosis*

Given that the pattern contains lexical information (the token `began`), we test if this information is also in the text fragment. Being so, there is a lexico-syntactic overlap between the pattern and each of the text segments. In Subsection 4.3 we will explain how we discard the first matched text fragment.

4.2 Running strong, inflected and weak patterns

After the matching, the generation of new questions (and extraction of their respective answers) is straightforward, given that we keep track of the Q/A pairs that originated each pattern and the links between them.

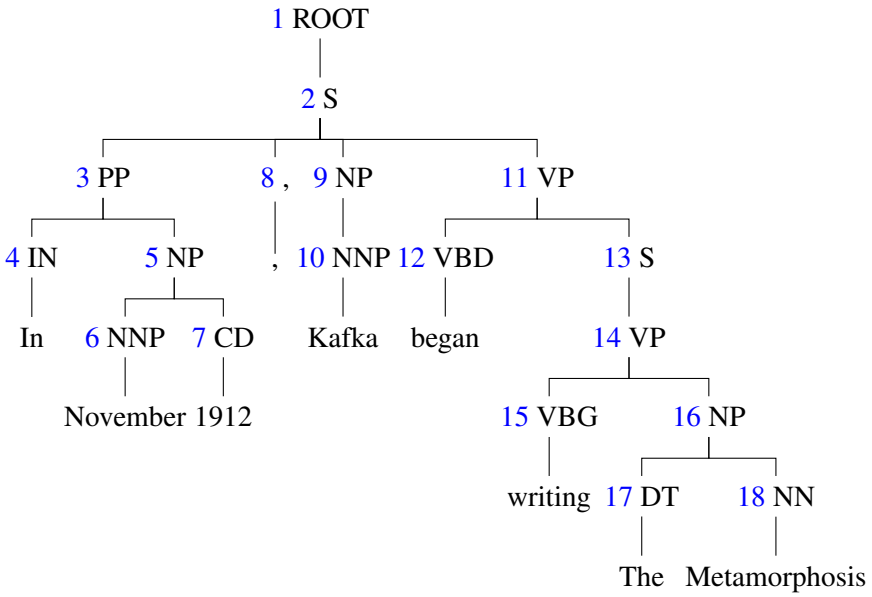


Figure 6: Parse tree of the sentence *In November 1912, Kafka began writing The Metamorphosis*.

The strategies for generating new Q/A pairs differ according to the type of pattern and go as follows:

`strong` patterns – There is a direct unification of the text fragment that matched a pattern with the syntactic constituents of the question that generated the pattern. The *Wh*-phrase from the seed question is used.

Figure 7 shows graphically the generation of a question for a `strong` pattern and all the information involved in this subtask. The colours identify components (either text or syntactic constituents) that are related. In THE-MENTOR, these associations are done by means of indices that relate the constituents of the syntactic structure of a question to a learned pattern. That is, a pattern $\texttt{NP}_2$ $\texttt{VBD}_1$ `by` NP{*ANSWER*} is augmented with indices that refer to the constituents of the seed question: WHNP_0 VBD_1 NP_2.

When the match between a pattern and a fragment in the target text occurs, we use those indices to build a new question, with the respective content in the correct position.

`inflected` patterns – The same strategy used for the `strong` patterns is applied. However, the verb is inflected with the tense and person existing in the seed question and the auxiliary in the question is also used.

`weak` patterns – There is a direct unification of the text fragment that matched a pattern with the syntactic constituents of the question that generated it.

For all the components that do not appear in the fragment, the components in the question are used.

The verbal information of `strong` and `inflected` patterns is then used to fill the missing elements of the `weak` pattern. That is, since `weak` patterns are learned by discarding verbal

phrases in the query submitted to the search engine, sometimes questions cannot be generated because a verbal phrase is needed. Therefore, `strong` and `inflected` patterns, which have the same seed pair in their origin as the `weak` pattern, will contribute with their verbal phrases.

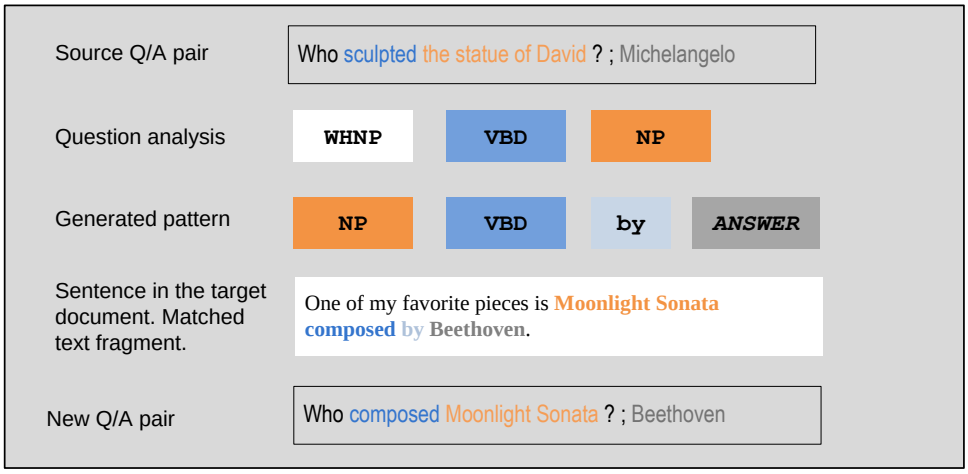


Figure 7: Question generation by THE-MENTOR.

4.3 Filtering

To discard low quality Q/A pairs, three different filters are applied:

Semantic Filter – A first filter forces the semantic match between the question and the answer, discarding the Q/A pairs where the answer does not comply with the question category. For that, at least one of words in the answer has to be associated with the category attributed to the question. By doing so, THE-MENTOR is able to rule out the incorrectly generated Q/A pair "*Who wrote The Metamorphosis?*; *November 1912*".

This filter is based on WordNet's (Fellbaum 1998) lexical hierarchy and on a set of fifty groups of WordNet synsets, each representing a question category, which we have manually created in a previous work (Silva et al. 2011). To find if a synset belongs to any of the pre-defined groups, we use a breadth-first search on the synset's hypernym tree. Thus, a word can be directly associated with a higher-level semantic concept, which represents a question category. For example, the category HUMAN:INDIVIDUAL is related with the synsets `person`, `individual`, `someone`, `somebody` and `mortal`. Given that *actor*, *leader* and *writer* are hyponyms of (at least) one of these synsets, all of these words are also associated with the category HUMAN:INDIVIDUAL.

This result is important to validate the following Q/A pair "*Who was François Rabelais?*; *An important 16th century writer*". Here, the answer agrees with the semantic category expected by the question (HUMAN:INDIVIDUAL) since the word *writer* is associated with that category. Therefore, this Q/A pair is not ruled out.

Search Filter – A different filter instantiates the permutation that generated the pattern using the elements of the extracted fragment. It is again sent to the search engine enclosed in double quotes. If a minimum number of results is returned, the question (and answer) is considered to be valid; if not, the fragment is filtered out.

Anaphora Filter – A final filter discards questions with anaphoric references, by using a set of regular expressions. Those which we have empirically verified that will not result in quality questions are also filtered out, for example *What is it?*, *Where is there?* or *What is one?*.

5. Evaluation

In this section we present a detailed evaluation of THE-MENTOR's main steps. We start by evaluating the pattern learning subtask and then we test THE-MENTOR in the QGSTEC 2010 development and test sets[10] and also in the Leonardo da Vinci Wikipedia page[11]. The former allows us to evaluate THE-MENTOR in a general domain; the latter allows the evaluation of THE-MENTOR in a specific domain – a text about art, as this domain is the ultimate target of THE-MENTOR. We conclude the section with a discussion about the attained results.

5.1 Pattern Learning

Regarding the pattern learning subtask, 139 factoid questions and their respective answers were used in our experiments (this same set was used in (Mendes et al. 2011)). Some of these were taken from an on-line trivia game, while others were hand crafted. Examples of such questions (with their respective answers) are "*In what city is the Louvre Museum?*; *Paris*", "*What is the capital of Russia?*; *Moscow*" or "*What year was Beethoven born?*; *1770*".

The set of Q/A pairs was automatically grouped, according to the process described in Section 3.1, resulting in 668 seed/validation pairs. Afterwards, the first Q/A pair of each seed/validation pair was submitted to Google (see Algorithm 1), and the 16 top ranked snippets were retrieved and used to learn the lexico-syntactic patterns' candidates. A total of 1348 patterns (399 `strong`, 729 `inflected` and 220 `weak`) were learned for 118 questions (from the original set of 139 questions).

The relation between the number of learned patterns (Y axis) and the number of existing questions for each category (X axis) is shown in Figure 8. As it can be seen, the quantity of the learned patterns is usually directly proportional to the number of seeds; however, there are some exceptions.

Table 3 makes the correspondence between the questions and their respective categories. It shows that the highest number of patterns were found for category HUMAN:INDIVIDUAL, which was the category with more seed/validation pairs. Nevertheless, we can also notice that the category ENTITY:LANGUAGE (*e.g.*, Which language is spoken in France?) had a small number of pairs (4) and led to a large number of patterns (195), all belonging to the type `strong` and `inflected`. Also, categories LOCATION:CITY and LOCATION:STATE, despite having a similar number of seed/validation pairs, gave rise to a very disparate number of learned patterns: 115 and 69 learned patterns for 11 and 12 questions, respectively.

The achieved results suggest that the pattern learning subtask depends not only on the quantity of questions given as seed, but also on their content (or quality). However, after analysing the

10. `http://questiongeneration.org/QG2010`.
11. `http://en.wikipedia.org/wiki/Leonardo_da_Vinci`.

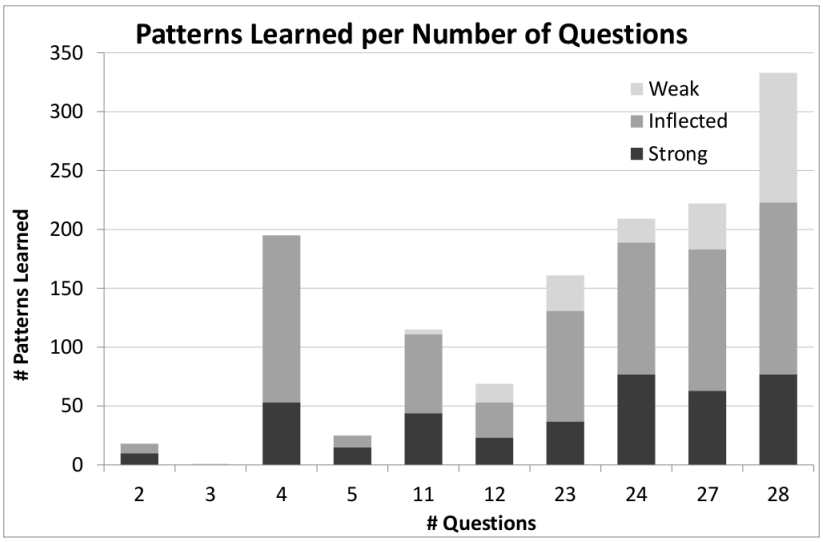


Figure 8: Number of patterns per number of questions (for each category).

		Learned Patterns			
# Questions	Category	`strong`	`inflected`	`weak`	Total
2	LOCATION:MOUNTAIN	10	8	0	18
3	ENTITY:SPORT	0	0	1	1
4	**ENTITY:LANGUAGE**	53	142	0	**195**
5	ENTITY:CURRENCY	15	10	0	25
11	LOCATION:CITY	44	67	4	115
12	LOCATION:STATE	23	30	16	69
23	NUMERIC:DATE	37	94	30	161
24	LOCATION:COUNTRY	77	112	20	209
27	LOCATION:OTHER	63	120	39	222
28	**HUMAN:INDIVIDUAL**	77	146	110	**333**
Total		399	729	220	1348

Table 3: Distribution of the number of questions per category by types of learned patterns.

original set of 139 questions, there is nothing in their syntax or semantics that can give us a clue about the reasons why some questions resulted in patterns and others did not. For instance, the Q/A pair "*In what country were the 1948 Summer Olympics held?*; *England*" resulted in a pattern; however, no pattern was learned for the Q/A pair "*In what country were the 1964 Summer Olympics held?*; *Japan*" and "*In what country were the 1992 Summer Olympics held?*; *Spain*". Note that all these three Q/A pairs were used as seed and validation (recall that THE-MENTOR creates 6 seed/validation pairs from 3 Q/A pairs).

The quality of a question is difficult to estimate, since very similar questions can trigger a very disparate number of patterns. The learning of patterns relies on the large number of occurrences of certain information in the information sources, which depends on several factors, like the characteristics of the used information sources and the actuality/notoriety of the entities and/or events in the question. For instance, our feeling is that it is probable that more patterns are learned if the question asks for the birthplace of Leonardo Da Vinci, rather than the birthplace of José Malhoa (a Portuguese painter), since the former is more well known than the latter. However, the achieved results are also influenced by the type of corpora used and this situation would probably not occur if the patterns were to be learned from corpora composed of documents about Portuguese art (instead of the Web). Finally, the algorithm we employ to learn patterns also depends on the documents the search engine considers relevant to the posed query (and the ranking algorithms used at the moment). Currently, this is a variable that we cannot control.

Despite our intuition about the desirable properties of a question and the used information sources to our pattern learning approach, this is certainly a topic that deserves further investigation and should be a direction of future work.

5.2 Question Generation

5.2.1 EVALUATION CRITERIA

In order to evaluate the questions[12] generated by THE-MENTOR, we follow the approach of Chen et al. (2009). These authors consider the questions to be *plausible* if they are grammatically correct and if they make sense regarding the text from which they were extracted; they are considered to be *implausible*, otherwise. However, and due to the fact that the system does not handle anaphora and can generate anaphoric questions (*e.g.*, *When was he sent to the front?*), we decided to add another criteria: if the question is plausible, but contains, for instance, a pronoun, and context is needed in order to understand it. In conclusion, in our evaluation, each question is marked with one of the following tags:

- PL: for plausible, non-anaphoric questions. That is, if the question is well formulated at the lexical, syntactical and semantical levels and makes sense in the context of the sentence that originate it. For instance *Who is the president of the Queen International Fan Club?* is a question marked as PL;

- IMPL: for implausible questions. That is, if the question is not well formulated in lexical, syntactical or semantical terms, or if it could not be inferred from the sentence that originate it. As examples, questions such as *Where was France invented?* and *Who is Linux?* are marked as IMPL.

- CTXPL: for plausible questions, being given a certain context. That is, if the sentence is well formulated in lexical, syntactical and semantical terms, but contains pronouns or other references that can only be understood if the user is aware of the context of the question. An example of a question marked as CTXPL is *Who lost his only son?*.

12. Although we have described THE-MENTOR's strategy to create new Q/A pairs, here we only evaluate the generated questions and ignore the appropriateness of the extracted answer.

5.2.2 OPEN DOMAIN QUESTION GENERATION

In a first evaluation we used the development corpus from QGSTEC 2010 task B (sentences), which contains 81 sentences extracted from Wikipedia, OpenLearn and YahooAnswers. Three experiments were carried out to evaluate the filtering process described in Section 4.3; experiments were also conducted taking into account the involved type of patterns. In a first experiment all the filters were applied – All Filters (Semantic Filter + Search Filter + Anaphora Filter) – which resulted in 29 generated questions. In the second experiment only the filter related with the question category were applied – Semantic Filter. This filtering added an extra set of 103 questions to the previous set of 29 questions. Finally, a third experiment ran with no filters – No Filters. In addition to the 29+103 questions, 1820 questions were generated. Nevertheless, for evaluation purposes, only the first 500 were evaluated. Results are shown in Table 4.[13]

All Filters (Total: 29)											
`strong` patterns				`inflected` patterns				`weak` patterns			
PL	IMPL	CTXPL	Total	PL	IMPL	CTXPL	Total	PL	IMPL	CTXPL	Total
8	10	3	21	**1**	1	0	2	**2**	4	0	6
Semantic Filter (New: 103)											
`strong` patterns				`inflected` patterns				`weak` patterns			
PL	IMPL	CTXPL	Total	PL	IMPL	CTXPL	Total	PL	IMPL	CTXPL	Total
4	6	1	11	**3**	3	1	7	**1**	81	3	85
No Filters (New: 1820, Evaluated: 500)											
`strong` patterns				`inflected` patterns				`weak` patterns			
PL	IMPL	CTXPL	Total	PL	IMPL	CTXPL	Total	PL	IMPL	CTXPL	Total
4	41	1	46	**2**	42	4	48	**3**	402	1	406
Total PL				Total IMPL				Total CTXPL			
28				**590**				**14**			

Table 4: Results from the QGSTEC 2010 corpus

The first conclusion we can take from these results is that the three approaches – All Filters, Semantic Filters and No Filters – have almost equally contributed to the final set of (28) plausible questions (All Filters (11), Semantic Filter (8) and No Filters (9)). As expected, `strong` and `inflected` patterns are more precise than `weak` patterns, and their use results in 36-50% of plausible questions, considering the All Filters and Semantic Filter approach (8/21 and 4/11 for `strong` patterns, 1/2 and 3/7 for `inflected` patterns). Moreover, and also as expected, when no filter is applied, `weak` patterns over-generate implausible questions (402 IMPL questions in 406).

Regarding the number of questions generated by each sentence, in 42 questions that were labelled as PL or CTXPL, 20 had the same source sentence. Thus, only 22 from the 81 sentences of QGSTEC 2010 resulted in PL or CTXPL sentences. Considering these 22 questions, 12 were generated from sentences from OpenLearn, 10 from the Wikipedia and 2 from YahooAnswers. As there were around 24 sentences from YahooAnswers and Wikipedia (and the rest was from Open-

13. To profit from the Tregex engine, we repeated this same experiment by mapping our patterns into Tregex format and the same set of questions were obtained.

Learn), YahooAnswers seems to be a less useful source for QG with THE-MENTOR. However, more experiments need to be carried out to corroborate this observation.

A final analysis was made on the distribution of the 29 questions generated with all filters regarding their categories. Results can be seen in Figure 9, where all the remaining categories had 0 questions associated and are not in the chart.

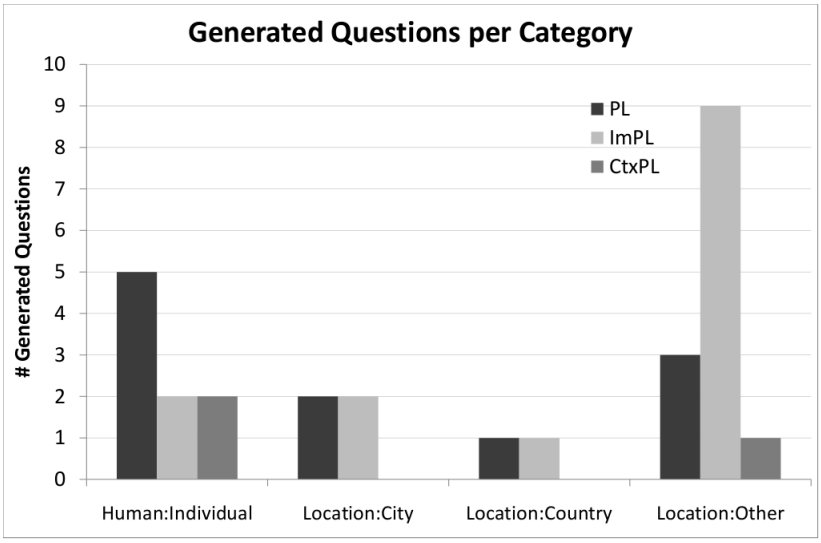


Figure 9: Evaluation of generated questions according to their category (all filters applied). Category LOCATION:STATE was omitted since only one IMPL question was generated from one `inflected` pattern.

Apparently, a category such as LOCATION:OTHER is too general, thus resulting in many implausible questions. More specific categories such as HUMAN:INDIVIDUAL, LOCATION:CITY or LOCATION:COUNTRY attain more balanced results between the number of plausible and implausible questions generated. Also, looking again at Table 3, all the categories that resulted in a low number of patterns did not produce any plausible question.

5.2.3 COMPARING THE-MENTOR WITH OTHER SYSTEMS

The previous experiments using the development corpus of QGSTEC 2010 permitted us to evaluate the system and to chose the setting where THE-MENTOR achieved the best results, that is, the setting that we would use if we were participating in the QGSTEC 2010 task B (sentences). However, it is not possible to fully simulate our participation in this challenge, since THE-MENTOR is not prepared to receive as input a sentence and the question types that should be generated (THE-MENTOR only works with the sentence from which to generate the questions). Therefore, our results are judged with only three parameters used in the challenge: (a) relevance, (b) syntactic correctness and fluency, and (c) ambiguity (more details about the evaluation guidelines can be found in Rus et al. (2010b)).

Our human evaluator started by studying the guidelines and the last twenty questions evaluated by QGSTEC 2010 evaluators. Then, our annotator evaluated the first fifty questions generated by the

participating systems, and the Cohen's kappa coefficient (Cohen 1960) was computed to calculate the agreement between our evaluator and the evaluators of QGSTEC 2010 task B (as their scores are available). Tables 5, 6 and 7 summarize these results for each of the evaluation parameters under consideration. It should be said that, as the QGSTEC scores are presented as the average between each of the evaluator's scores, a normalization of values was made. The first column expresses our evaluation and the first row the QGSTEC 2010 scores (for instance, considering the first cell, it means that our evaluator gave score 1 and their evaluators gave score 1 or 1.5 to 39 questions).

Relevance	1 OR 1.5	2 OR 2.5	3 OR 3.5	4	Total
1	**39**	0	0	0	39
2	3	**2**	3	0	5
3	3	2	**1**	0	5
4	0	0	1	**0**	1
Total	45	4	1	0	**50**

Table 5: Inter-annotator agreement for the relevance parameter.

Syntax	1 OR 1.5	2 OR 2.5	3 OR 3.5	4	Total
1	**15**	0	0	0	15
2	9	**10**	0	0	19
3	0	4	**10**	0	14
4	0	0	1	**2**	2
Total	24	14	10	2	**50**

Table 6: Inter-annotator agreement for the syntactic correctness and fluency parameter.

Ambiguity	1 OR 1.5	2 OR 2.5	4	Total
1	**41**	0	0	41
2	3	**2**	0	5
3	0	2	**2**	4
Total	16	21	4	**50**

Table 7: Inter-annotator agreement for the ambiguity parameter.

Regarding relevance, the inter-annotator agreement is considered fair (0.38); in what concerns the other two variables, they are considered substantial (respectively, 0.62 and 0.65).

Afterwards, our annotator evaluated the questions generated by THE-MENTOR in the sentences set. It generated 25 questions using our best predicted setting, that is, it ran with `strong` and `inflected` patterns and with the semantic filter. Table 8 shows the scores attributed by our annotator to the questions generated by THE-MENTOR according to each one of the parameters. Recall that lower scores are better.

	RELEVANCE	SYNTAX	AMBIGUITY
#1	21	12	15
#2	1	3	4
#3	3	6	6
#4	0	4	n.a.
Total	25	25	25

Table 8: Results achieved by THE-MENTOR.

Table 9 shows the comparison between the results achieved by THE-MENTOR and the results achieved by the other systems participating at QGSTEC 2010.[14]

System	RELEVANCE	SYNTAX	AMBIGUITY	Good Questions
A	1.61 ± 0.74	2.06 ± 1.01	1.52 ± 0.63	181/354 (51%)
B	1.17 ± 0.48	1.75 ± 0.74	1.30 ± 0.41	122/165 (74%)
C	1.68 ± 0.87	2.44 ± 1.06	1.76 ± 0.69	83/209 (40%)
D	1.74 ± 0.99	2.64 ± 0.96	1.96 ± 0.71	44/168 (26%)
THE-MENTOR	1.28 ± 0.68	2.08 ± 1.88	1.64 ± 0.86	14/25 (56%)

Table 9: Comparison between the results achieved by THE-MENTOR and the other participating systems of QGSTEC 2010.

Results are similar if we compare relevance, syntactic correctness and fluency and ambiguity. Nevertheless, our results are clearly worse than other systems if we compare the number of questions generated. However, as said before, THE-MENTOR is not prepared to generate questions of different types for each sentence, thus the number of questions generated is low.

5.2.4 IN-DOMAIN QUESTION GENERATION

We also tested THE-MENTOR in a text about art, since this is the application domain of FalaComigo. We have chosen the Leonardo da Vinci page from Wikipedia, with 365 sentences.[15] This time, motivated by the previously achieved results, we decided not to use `weak` patterns. Once again, plausible questions such as *Where is Leonardo's earliest known dated work?* were marked as IMPL, if the sentences from where they were generated do not contain reference to its answer (*Leonardo's earliest known dated work is a drawing in pen and ink of the Arno valley, drawn on August 5, 1473*). However, we marked as PL questions that did not have an explicit answer in the sentence responsible for their generation, but it could be inferred from the sentence. An example is the question *In which country was Leonardo educated?*, as Italy can be inferred from *Born the illegitimate son of a notary, Piero da Vinci, and a peasant woman, Caterina, at Vinci in the region of Florence, Leonardo was educated in the studio of the renowned Florentine painter, Verrocchio*. The question *Who created*

14. For all systems, the number of good questions is calculated by taking into account only the three mentioned parameters.
15. This includes section titles and image captions.

the cartoon of the Virgin? was also marked as IMPL, because it lacks content (the name of the cartoon is *The Virgin and Child*). Finally, we marked as IMPL questions with grammatical errors, like *What sustain the canal during all seasons?*.

Since there was almost no difference between the questions generated when all the filters were applied and questions generated with the Semantic Filter (the latter contributed with one additional question, that was marked as IMPL), this filtering process was removed from the results presented in Table 10.

All Filters (Total: 41)							
`strong` patterns				`inflected` patterns			
PL	IMPL	CTXPL	Total	PL	IMPL	CTXPL	Total
14	13	4	31	**2**	8	0	10
No Filters (Total: 219)							
`strong` patterns				`inflected` patterns			
PL	IMPL	CTXPL	Total	PL	IMPL	CTXPL	Total
25	40	4	69	**21**	121	7	149

Total PL	Total IMPL	Total CTXPL	Total
62	**182**	**15**	**259**

Table 10: Results from Leonardo da Vinci's Wikipedia page.

The number of generated questions per sentence in this experiment is slightly smaller than the number of generated questions per sentence in the previous experiment. This time 259 questions were generated from 365 sentences and previously there were 135 questions generated (ignoring those originated in `weak` patterns) from 81 sentences: that is, a ratio of 1.4 and 1.6 generated questions per sentence, respectively. In contrast, the percentage of questions marked as PL or CTXPL was slightly higher with Leonardo da Vinci's page: 77 questions in 259, instead of 32 in 135. Thus, the percentage of successfully generated questions is between 24% and 30%, if `weak` patterns are ignored.

5.2.5 INTER-ANNOTATOR AGREEMENT

Two human annotators evaluated the questions generated from the Leonardo Da Vinci page, having THE-MENTOR ran with all the filters enabled. The annotators were not experts and had no previous training. The guidelines were defined and explained before the annotation and the annotators did not interact during this process. Again, the Cohen's kappa coefficient (Cohen 1960) was used to calculate the inter-annotator agreement.

Table 11 summarizes the inter-annotator agreement results.

Annotators agreed in the evaluation of 37 of the 41 questions: 15 PL, 21 IMPL and 1 CTXPL. There was no agreement in 4 questions. The question *Who was "Lionardo di ser Piero da Vinci"?*, generated from *his full birth name was "Lionardo di ser Piero da Vinci", meaning "Leonardo, (son) of (Mes)ser Piero from Vinci"* was classified as PL by one annotator and as IMPL by the other. This is an example of a question well formulated at the lexical, syntactic and semantic level and, according to an annotator, it could be posed to a user of THE-MENTOR; on the contrary, the other annotator considered the nonexistence of support from the sentence where it was generated as

	PL	IMPL	CTXPL	Total
PL	**15**	0	0	15
IMPL	1	**21**	3	25
CTXPL	0	0	**1**	1
Total	16	21	4	**41**

Table 11: Inter-annotator agreement.

sufficient reason to classify it as implausible (following the guidelines). The remaining 3 questions were classified as IMPL by one annotator and as CTXPL by the other. One example of such questions is *Who wrote an often-quoted letter?* generated from *At this time Leonardo wrote an often-quoted letter to Ludovico, describing the many marvellous and diverse things that he could achieve in the field of engineering and informing the Lord that he could also paint.*

Therefore, the relative observed agreement among raters, Pr(a), is 0.90, the probability of chance agreement, Pr(e), is 0.46 and, finally, the kappa coefficient (K)[16] is 0.82, which is, as considered by some authors an almost perfect agreement.

The guidelines for this evaluation were set ahead. Our main goals during the definition of the guidelines were to make them simple and intuitive (to minimize uncertainty during the annotation process), general enough to be used in other evaluations and to allow comparisons between QG systems, but also adapted to the expected output of THE-MENTOR (for example, given that the system does not solve anaphora, it generates anaphoric questions and in those situations the tag CTXPL is available). In our opinion, these characteristics of the guidelines led to the high agreement between annotators. Also, the evaluation of certain questions was a straightforward process that simply did not raise any doubts: for instance, it was obvious to mark as IMPL the question *In which continent is a drawing*, taken from *Leonardo's earliest known dated work is a drawing in pen and ink of the Arno valley, drawn on August 5, 1473*. However, it is also visible the different perceptions about the quality of the generated questions by the different annotators. Whereas one annotator classified the question by itself and its usefulness when presented alone to a human user, the other classified it taking in consideration the entire task of QG: for instance, despite the lexical, syntactic and semantic correction of the generated question, if the answer can not be found in the supporting sentence, the annotator marked it as implausible.

6. Conclusions and Future Work

We described the QG task as performed by THE-MENTOR, a platform that automatically generates multiple-choice tests. THE-MENTOR takes as input a set of Q/A pairs, constructs seeds and learns lexico-syntactic patterns from these seeds. Three types of patterns are possible: `strong`, `inflected` and `weak` patterns. The first type imposes more constraints, since the seed elements need to be present in the learned patterns; the second adds some flexibility, allowing seeds and patterns not to agree in the verb forms; the third only imposes noun-phrases from the seeds to be present in the patterns. These patterns are used to generate questions, being given a text as input. The filters applied both to the learned patterns and to the generated questions were also described. Moreover,

16. K = (Pr(a)-Pr(e))/(1-Pr(e)).

we have evaluated THE-MENTOR on the QGSTEC 2010 corpus, as well as on a Wikipedia page. Results show that:

- There is a linear relation between the number of seeds with a certain category and the number of resulting patterns, although there are exceptions;
- Only questions with a large amount of patterns contribute to a successful QG;
- Questions with coarse-grained categories mostly generate implausible questions;
- `Weak` patterns over-generate implausible questions;
- The percentage of successfully generated questions is between 24% and 30%, if `weak` patterns are ignored;
- Simulating the evaluation of THE-MENTOR with the QGSTEC 2010 test set, we managed to obtain average results in what concerns relevance, syntactic correctness and fluency, and ambiguity, but THE-MENTOR generates significantly fewer questions than the other systems.

There is still plenty of room for improvement in THE-MENTOR. However, there are two points worth mentioning regarding the achieved results. On one hand, some questions were marked as IMPL but could be easily converted to PL questions: for instance, the question *Who is Linux?* could be transformed into *What is Linux?*. On the other hand, some questions were marked as IMPL uniquely because the answer did not appear in the sentence that originated them (for instance *Who invented the Hubble Space Telescope?*). However, it could be worth presenting such questions to the user of THE-MENTOR, depending on the application goal (for instance, if THE-MENTOR is used to help a tutor on the creation of tests). In this case, the judgement of the correctness of the user's answer has to be done by other means, either by a human evaluator (*e.g.*, the tutor) or by using other strategies.

Regarding future work directions, we have identified a set of research points that will be the target of our next efforts, namely:

- Given the co-references that usually appear in a text which, in addition to complex syntactic constructions, make it harder to process, we intend to follow the approaches of Heilman and Smith (2010) and Kalady et al. (2010) and pre-process the texts used to learn patterns and generate the question/answer pairs. With this, we hope to obtain more plausible questions.
- One of the main causes for obtaining such an amount of implausible questions is the use of a coarse-grained question classification. We have seen that questions classified as LOCATION:OTHER originate a large number of implausible questions, which is not the case of questions classified as LOCATION:COUNTRY or LOCATION:CITY. Thus, this taxonomy needs to be altered and enriched with more fine-grained categories.
- We will also add more questions to certain categories and see if this extra set can lead to plausible questions of that category.
- We will move from factoid to definition questions, although this will probably led us to different strategies.

- The learning process can become more flexible if we allow the use of synonyms in the queries submitted to the search engine. In fact, by allowing different verb forms we have created the `inflected` patterns that have significantly contributed to the QG of plausible questions. Thus, a next step will be to expand queries during the learning process.
- Still considering the learning process, the validation step needs to be revised, as some of the patterns that are considered to be too specific, could be a good source of questions. For instance, a sentence such as *the Statue of David was sculpted around 1504 by Michelangelo* will never originate a pattern to a question such as *Who sculpted the Statue of David?* because of the specificity of the appearing date. However, if we are able to tag 1504 as a DATE, that sentence could originate a pattern, that would be able to trigger questions from sentences like *Mount Rushmore National Memorial was sculpted around 1935 by Gutzon Borglum*. That is, not only syntactic categories and tokens will be taken into account in the pattern learning process, but named entities should also be considered.
- We will also follow Heilman and Smith (2009) ideas and re-rank our questions. In fact, even if we consider the hypothesis of presenting a user with 100 questions from which 80 are implausible, it is better to have the most probable plausible questions in the first positions.
- Finally, we are currently porting THE-MENTOR to Portuguese and analysing the possibility of using dependency grammars, as they will allow the relation of long distance constituents.

Acknowledgments

This work was supported by FCT (INESC-ID multiannual funding) through the PIDDAC Program funds, and also through the project FALACOMIGO (ProjectoVII em co-promoção, QREN n 13449) that supports Sérgio Curto's fellowship. Ana Cristina Mendes is supported by a PhD fellowship from Fundação para a Ciência e a Tecnologia (SFRH/BD/43487/2008).

References

Elizabeth S. Adams, Linda Carswell, Amruth Kumar, Jeanine Meyer, Ainslie Ellis, Patrick Hall, and John Motil. Interactive multimedia pedagogies: report of the working group on interactive multimedia pedagogy. *SIGCSE Bull.*, 28(SI):182–191, 1996.

Lee Becker, Rodney D. Nielsen, and Wayne H. Ward. What a pilot study says about running a question generation challenge. In *The 2nd Workshop on Question Generation*, 2009.

Niels Ole Bernsen and Laila Dybkjær. Domain-Oriented Conversation with H.C. Andersen. In *Proc. Workshop on Affective Dialogue Systems*, pages 142–153. Springer, 2004.

Phil Blunsom, Krystle Kocik, and James R. Curran. Question classification with log-linear models. In *Proc. 29th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '06)*, pages 615–616. ACM, 2006.

Kristy Elizabeth Boyer, William Lahti, Robert Phillips, Michael Wallis, Mladen Vouk, and James Lester. An empirically derived question taxonomy for task oriented tutorial dialogue. In *The 2nd Workshop on Question Generation*, 2009.

Eric Brill, Susan Dumais, and Michele Banko. An analysis of the AskMSR question-answering system. In *Proc. ACL-02 conference on Empirical methods in natural language processing*, EMNLP '02, pages 257–264. Association for Computational Linguistics, 2002.

Wei Chen, Gregory Aist, , and Jack Mostow. Generating questions automatically from informational text. In *The 2nd Workshop on Question Generation*, 2009.

Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37, 1960.

Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

Alejandro G. Figueroa and Günter Neumann. Genetic algorithms for data-driven web question answering. *Evol. Comput.*, 16(1):89–125, 2008.

Corina Forăscu and Iuliana Drăghici. Question generation: Taxonomies and data. In *The 2nd Workshop on Question Generation*, 2009.

Michael Heilman and Noah Smith. Ranking automatically generated questions as a shared task. In *The 2nd Workshop on Question Generation*, 2009.

Michael Heilman and Noah Smith. Extracting simplified statements for factual question generation. In *The 3rd Workshop on Question Generation*, 2010.

Zhiheng Huang, Marcus Thint, and Zengchang Qin. Question classification using head words and their hypernyms. In *Proc. Conference on Empirical Methods in Natural Language Processing*, pages 927–936, 2008.

Kateryna Ignatova, Delphine Bernhard, and Iryna Gurevych. Generating high quality questions from low quality questions. In *Proc. Workshop on the Question Generation Shared Task and Evaluation Challenge*, 2008.

John Judge, Aoife Cahill, and Josef van Genabith. Questionbank: creating a corpus of parse-annotated questions. In *Proc. 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics (ACL-44)*, pages 497–504. Association for Computational Linguistics, 2006.

Saidalavi Kalady, Ajeesh Elikkottil, and Rajarshi Das. Natural language question generation using syntax and keywords. In *The 3rd Workshop on Question Generation*, 2010.

Nikiforos Karamanis, Le An Ha, and Ruslan Mitkov. Generating multiple-choice test items from medical text: a pilot study. In *Proc. Fourth International Natural Language Generation Conference (INLG '06)*, pages 111–113. Association for Computational Linguistics, 2006.

Stefan Kopp, Lars Gesellensetter, Nicole Krämer, and Ipke Wachsmuth. A conversational agent as museum guide: design and evaluation of a real-world application. In *Proc. 5th International Working Conference on Intelligent Virtual Agents*, pages 329–343, 2005.

Anton Leuski, Ronakkumar Patel, and David Traum. Building effective question answering characters. In *Proc. 7th SIGdial Workshop on Discourse and Dialogue*, pages 18–27, 2006.

Roger Levy and Galen Andrew. Tregex and Tsurgeon: tools for querying and manipulating tree data structures. In *Proc. 5th international conference on Language Resources and Evaluation*, 2006.

Xin Li and Dan Roth. Learning question classifiers. In *Proc. 19th international conference on Computational linguistics*, pages 1–7. Association for Computational Linguistics, 2002.

Filipe Martins, Ana Mendes, Márcio Viveiros, Joana Paulo Pardal, Pedro Arez, Nuno J. Mamede, and João Paulo Neto. Reengineering a domain-independent framework for spoken dialogue systems. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing, an ACL 2008 Workshop*, pages 68–76. Workshop of ACL, 2008.

Ana Cristina Mendes, Rui Prada, and Luísa Coheur. Adapting a virtual agent to users' vocabulary and needs. In *Proc. 9th International Conference on Intelligent Virtual Agents*, Lecture Notes in Artificial Intelligence, pages 529–530. Springer-Verlag, 2009.

Ana Cristina Mendes, Sérgio Curto, and Luísa Coheur. Bootstrapping Multiple-Choice Tests with The-MENTOR. In *Proc. 12th International Conference on Intelligent Text Processing and Computational Linguistics (CICling)*, pages 451–462, 2011.

Ruslan Mitkov, Le An Ha, and Nikiforos Karamanis. A computer-aided environment for generating multiple-choice test items. *Nat. Lang. Eng.*, 12(2):177–194, 2006.

Catarina Moreira, Ana Cristina Mendes, Luísa Coheur, and Bruno Martins. Towards the rapid development of a natural language understanding module. In *Proc. 10th international conference on Intelligent virtual agents*, pages 309–315. Springer-Verlag, 2011.

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; Proc. Main Conference*, pages 404–411. Association for Computational Linguistics, 2007.

Paul Piwek and Svetlana Stoyanchen. Question generation in the coda project. In *The 3rd Workshop on Question Generation*, 2010.

Deepak Ravichandran and Eduard Hovy. Learning surface text patterns for a question answering system. In *Proc. 40th Annual Meeting on Association for Computational Linguistics (ACL '02)*, pages 41–47. Association for Computational Linguistics, 2002.

Vasile Rus, Brendan Wyse, Paul Piwek, Mihai Lintean, Svetlana Soyanchev, and Cristian Moldovan. The first question generation shared task evaluation challenge. In *Proc. 6th international Natural Language Generation Conference*, pages 251–257, 2010a.

Vasile Rus, Brendan Wyse, Paul Piwek, Mihai Lintean, Svetlana Soyanchev, and Cristian Moldovan. Overview of the first question generation shared task evaluation challenge. In *Proc. 3rd Workshop on Question Generation*, 2010b.

João Silva, Luísa Coheur, Ana Mendes, and Andreas Wichert. From symbolic to sub-symbolic information in question classification. *Artificial Intelligence Review*, 35:137–154, 2011.

Martin M. Soubbotin. Patterns of potential answer expressions as clues to the right answers. In *Proc. 10th Text REtrieval Conference*, pages 293–302, 2001.

Brendan Wyse and Paul Piwek. Generating questions from openlearn study units. In *The 2nd Workshop on Question Generation*, 2009.

Feiyu Xu, Hans Uszkoreit, Sebastian Krause, and Hong Li. Boosting relation extraction with limited closed-world knowledge. In *Proc. 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pages 1354–1362. Association for Computational Linguistics, 2010.

Dell Zhang and Wee Sun Lee. Question classification using support vector machines. In *Proc. 26th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 26–32. ACM, 2003.