Automated EA-type Question Generation from Annotated Texts

László Bednarik, László Kovács
University of Miskolc/Department of Information Technology, Hungary
laszlobednarik@gmail.com, kovacs@iit.uni-miskolc.hu

Abstract—The field of automated question generation is an active investigated area within the development of e-learning systems. The paper gives first an overview of the main methods and then the architecture of the proposed system is presented in details. The novelty of the proposed application lies in the target language Hungarian and in the application of soft computing methods for generation of semi-synonyms as no free thesaurus is available for the target language. The last part of the paper shows an example process of test generation.

I. INTRODUCTION

The usage and importance of e-learning systems is increasing in current internet based society. The first e-learning systems could be used only for passive presentation of information with fixed material and content. As the learners have different abilities and needs, the fixed content yields in a low efficiency. In order to improve the quality of teaching, the up-to-date commercial or free public educational software products can provide a high level of interactivity and personalization. An in important component of the e-learning tools is the test module having different roles within the system. First, the module is used be the users to measure the achieved knowledge level. On the other hand, it can be used as feedback to the teachers and developers on how successful the training material and method is.

The current commercial and free e-learning systems like Ilias, Moodle contain a sophisticated test module with different question types. As these questions are created with manual work by the tutors, the question pools consist of only limited number of questions and the extension of the pool is a relative expensive task. Based on this reasons, there is a need in the community on automated test generation. There are several attempts and approaches of automated question generation; it is an active research area. In the literature, the first larger wave of works on computerized tutorials and question generation appeared around 2000. Before that time, the related researches focused on the semantic aspects of question generation like which can be used as a good methodological base of the automated systems. In the last decades the appearing works investigate the efficient integration of the different related methods as semantic or linguistic knowledge base.

Although now days there are many successful solutions on the field of automated question generation (AQG), the application of these methods have still some limitations. One of the key problems is that these methods are bounded to specific dominating languages because of the applied grammar module. On the other hand they use mainly such semantic knowledge bases which are limited only to given languages and problem domains. Thus the main goal of this project was to develop a more flexible and open framework for the Hungarian language. The other focus of the work is the investigation how effective is the application of general soft computing stochastic methods in the processing of semantic and grammatical problem domains. In the current status, the system capability is restricted to a single type of questions, namely the fill-in type questions.

The paper gives first an overview of the main methods of automated question generation than the architecture of the proposed system is presented. The chapter after architecture description presents the applied clustering algorithm to determine the semi-synonyms of the words. The last part of the paper shows an example process of test generation.

II. REVIEW OF AQG METHODS

The first attempts on automated question generation (AQG) methods go back to 1990’s. The AQG tools can be considered as the integration of several tools from related areas of text mining, natural language processing, semantic knowledge management, pedagogy and software development. Considering the methodological part, a pioneer work on comprehensive analysis of question types is given in [13], where the following complexity levels are distinguished: simple recall, comprehension, application,
analysis, synthesis and evaluation. On the formal level, the question can be presented in different forms.

Regarding the applied internal algorithm, a widely used method is the rule-based implementation. In this case, there are formulas pre-defined by human experts for sentences. If a sentence meets one of these formulas, it can be converted into given sentences using a pre-defined conversion formula. The second type of algorithms uses some learning algorithms to detect the conversion rule and to learn the importance of the different input sentences.

A base approach on AQG is the proposal of [14] where six different question types against the Wordnet knowledge base are defined as follows: definition, synonym, antonym, hypernym, hyponym and fill-in type (choice). The question of synonym type requires the retrieval of synonyms; in case of the antonym type, the words of opposite meaning are retrieved. The concepts hypernym and hyponym corresponds to the specialization and generalization relationships. In the work of [15], 24 different question types are defined where also the frequency of the types are analyzed. The most important question types are here the hint, procedural and simple recall and confirmation types. Based on the complexity, the choice type was selected for implementation from several reasons. The main reason is that it does not need to generate new sentences; the original sentence can be used in the test. The generation of new sentences is a more complex task requiring the full command of the grammar and the knowledge of more general semantic. The difficulty of fill-in type questions lies in the election of the options. The incorrect option words should be not substitutable with the original word in the given context. On the other hand it should be not far from it in the meaning in order to set the required difficulty level.

Considering the QAG application, there are several approaches in the literature on the implementation architecture. One of the first descriptions of the base framework for AQG system is presented in [16]. The engine consists of three main parts:

- sentence extraction using a preference learning method
- estimation of the blank positions with a conditional random field approach
- generation of candidate words for the blank positions with statistical pattern matching method
- reduction of the set of candidate words

During the preference learning phase, the structure and word set of previous similar topics are analyzed and questions of the same kind are generated. For the efficient generation of preference learning a Ranking Voted Perception algorithm was introduced in [17].

In the work of [18], the module consists of three main steps: sentence extraction, determining the blank part and generation optional words. The selection of sentences, blank positions and optional words are determined with help of machine learning methods using statistical and discriminative models. In the approach of [19] the word for blank positions are restricted to a given grammatical role (for example noun). The words for options are generated with the usage of the WordNet semantic knowledge base.

OpenLearn is a recently developed open framework which can be used as an online educational repository for development of AQ applications. The framework uses a specific XML format for storage of teaching materials. This XML format can provide a large freedom in application of specific annotations. The CEIST [20] method developed for this framework has a preprocessing phase where the document content given in pure text format is converted into parsing tree structure by means of NL grammar parser module. The main unit of the engine, a pattern matching algorithm, is used to find pre-defined patterns in the tree. For the sentences found in the tree questions are generated using a schema of conversion rules. The outline of the proposed architecture is shown in Fig 1.

### III. STRUCTURE OF THE PROPOSED SYSTEM

One of the main goals of the development was to provide an open architecture containing only freely available tools and self-developed components for processing of documents in Hungarian language. This restriction means that no WordNet-like module and no high level language grammar parser can be included in the architecture as currently none of them is freely available in the community for the
Hungarian language. From this reason, the proposed system is designed only for processing of a single question type, namely of fill-in questions. Another consequence of the restriction is that the semantic analysis uses a self-training soft-computing method instead of application of existing powerful ontology knowledge base.

On the highest architecture level, the proposed engine has a lot of common elements with other approaches; it consists of the following modules:

- text preprocessing
  - annotations
  - stemming
- domain preprocessing:
  - topic based clustering of words (semi-synonyms)
  - generation of term hierarchy
- selection of key word: classification of words
- selection of the candidate words using the word clusters
  - similar words
  - un-similar words
- grammatical alignment of the candidate words
- generation of test
- evaluation module.

The system uses a semi-automated selection method for determining the base sentences of the questions. Initially, an annotation framework is used to denote by human experts which sentences are suitable and optimal for question generation. Having a large set of training examples, a classification method is applied to find out the hidden relationship between the sentence parameters and the experts’ decisions. Like in the selection of candidate words for the question, also here a CPN (Counter Propagation Network) variant was invoked. Within the text preprocessing phase, another important step is the grammatical preprocessing of the words. As no powerful commercial language morpheme analyzer could be involved, a simplified free version of the Szószablya framework [21] is applied here. This module can determine the word class and stem of the words, but the dictionary of the free version can’t cover all words of the incoming documents.

In the source document each word is extended with a set of attributes including

- stem
- word class
- position within the sentence
- neighboring words
- semantic cluster.

The clustering module is used to determine the semi-synonyms of the words. The distance between two words is defined as a semantic distance based on the similarity of the neighbor sets.

$$d(i, j) = \frac{f_{ij}}{\max\{f_i, f_j\}}$$

where, $f_i$ denotes the number of sentences containing the word $w_i$ and $f_{ij}$ is the number of sentences containing both $w_i$ and $w_j$. The words assigned to the same cluster are considered as synonyms as they are substitutable with each others.

After the selection of the sentence for question generation, the blank position within the sentence is selected using a classification algorithm based on the CPN neural network.

IV. SEMANTIC CLUSTERING OF THE WORDS

Although, the literature on clustering is very rich, there is an aspect that did not gain much interest in the recent years. The aspect in the focus is the similarity threshold within a cluster. Based on the informal definition of clustering, the elements within a cluster should be more similar to each other than the elements of different clusters.

The goal of the investigation is to find clustering methods preserving the similarity requirements. In order to meet this requirement, the following two constraints are defined on the clustering model:

- for every object pair with a smaller distance than a threshold, there exists a cluster containing both elements of the pair;
- for every object pair with a larger distance than a threshold, there is no cluster containing both elements of the pair.

The first criterion ensures that examining the target clusters, all object pairs similar to each other can be found. The second constraint says that a cluster does not contain dissimilar objects, it contains only similar objects.
Although there are many different clustering algorithms, the radius of the cluster is investigated only in the BIRCH and QT clustering algorithms. In BIRCH (balanced iterative reducing and clustering using hierarchies) [10], the radius of a cluster is defined as the average distance to the mean position. If the radius is greater than a threshold, the cluster is split into two subparts. In this approach the radius is defined as an average distance, thus the distance between two elements within the cluster may be larger than R. In this sense, the BIRCH algorithm defines a weak threshold on the volume of the cluster.

A much stronger constraint is given in the QT (Quality Threshold) clustering algorithm. The QT-clustering [11] uses two input parameters: first parameter is the maximum distance diameter and the second is the minimum cluster size. The diameter is defined on the following way:

\[
d = \max_{i,j} \sqrt{(x_i - x_j)^2}
\]

The size of cluster denotes here the number of elements within the cluster. The main steps of the QT-clustering are the followings:

1. Generating candidate clusters for each element where the candidate cluster is built up with a greedy algorithm. Taking an element D, the cluster contains all elements closer to D than the maximum radius.
2. The candidate cluster with the maximum size is selected as true cluster. The elements of this cluster are removed from the pool, the membership of the remaining candidate clusters are updated.
3. If the largest remaining cluster has a greater size than the minimum limit, go to step otherwise terminate the algorithm.

The QT algorithm generates non-overlapping clusters where some elements remain outside of clusters as outliers. The output of the clustering is a set of clusters of limited diameter, thus the similarity values between the elements are above a given threshold.

To find the optimal sequence of merge operations at a given threshold value is a combinatorial optimization problem. In order to find an appropriate heuristic solution, different approaches were tested.

Clusters whose distance is the smallest are joined. Every case when the distance between a cluster and several other clusters is equally the smallest results in a set-union in the clustering procedure. In order to increase the speed of the algorithm in addition to a breadth-first-search [5] a depth limited depth-first-search [5] algorithm has also been implemented.

Considering the cost of depth search in each expansion \( b \) (branching factor) new nodes arose and in a state space whose maximum depth is \( m \) the searching process needs a storage place \( \mathcal{O}(b^m) \) which means that the complexity of the place is linear. The requirement for storage place is very moderate as the path from a root node to a leaf node is needed to be stored completed with the undeveloped nodes next to each node in the path.

The time requirement of the search is \( \mathcal{O}(b^m) \) which is equal to the time requested in the worst case of the breadth search [6].

The best-first-search estimates the cost of reaching the aim from a state \( n \) with a heuristic evaluating function and it steps into a state where this cost is the lowest. The cost of storage place and time requirement of the search is \( \mathcal{O}(b^m) \) where \( m \) is the maximum depth of the search space. Complexity can considerably be reduced by choosing the heuristic function well. The degree of the reduction depends on the given problem and the quality of the heuristic reduction Fig. 2 [6].

![Figure 2. Comparison of the algorithms](image)

V. TEST RESULTS

The correctness of the model was proven by software implemented in Java language. Beyond testing the functional procedure the theoretical model we enabled the software to represent questions in electronic or printable
format. In the case of electronic version both the generated questions and possible answers appear in computer environment in front of the user. In this version user filling in the test can have a local menu on the screen offering all the possible answers to questions by clicking on the blank part of the sentence containing the question to be answered. After giving the answer the chosen alternative is automatically substituted in the sentence. For representing the test the filled test sheet can be saved in a file format and can be forwarded to the reviewer of the test. For representing the test sheet in a printable format the software indicates the place (blank part) of the word taken out as a question with dots for the person who is filling it in and the possible answers appear listed next to each other below sentences.

14. Database management system is a program system whose task is to provide connection to the .......... and to perform the task of internal maintenance.
1.) connection  2.) initiator 3.) information  4.) task  5.) database

In order to fill the test in a printed format the user has to underline the word judged to be appropriate or write it where the dots are. Fig. 3 shows an example for the test sheet in the form it appears.

Figure 3. Printable version of the automatically generated test sheet

ACKNOWLEDGMENT

This research was carried out as part of the TAMOP-4.2.1.B-10/2/KONV-2010-0001 project with support by the European Union, co-financed by the European Social Fund

REFERENCES