

Refined Distractor Generation with LSA and Stylometry for Automated Multiple Choice Question Generation

Josef Robert Moser^{1,3}, Christian Gütl^{1,2}, and Wei Liu³

¹ Institute for Information Systems and Computer Media
Graz University of Technology, Austria

² Business School, Curtin University, Australia

³ School of Computer Science and Software Engineering
The University of Western Australia, Australia

josef.robert.moser@gmail.com,
Christian.Guetl@iicm.tugraz.at,
wei@csse.uwa.edu.au

Abstract. As lifelong learning becomes increasingly important in our society, mechanisms allowing students to evaluate their progress must be provided. A commonly used and widely accepted feedback mechanism is the multiple-choice test. Manual creation of multiple choice questions is often a time consuming process involving many iterations of trial and error. Using text processing and natural language processing techniques, automated multiple choice question generation, in recent years, is getting much closer to reality than ever. However, one of the most difficult tasks in both manual creation and automated generation of this kind of tests is the creation of distractors, because unsuitable distractors allow students to easily guess the correct answer, which counteracts the goal of these questions. In this paper, we investigated the desired properties of distractors and identified relevant text processing algorithms, specifically, latent semantic analysis and stylometry, for distractor selection. The refined distractors are compared with baseline distractors generated by our existing Automated Question Creator (AQC). Our preliminary evaluation shows that this novel combined approach produces distractors with a higher quality than those of the baseline AQC system.

Keywords: Automated Multiple-Choice Question Generation, Distractors, Text Processing, Latent Semantic Analysis, Stylometry.

1 Introduction

Our globalised world demands continuous adaption of knowledge and skills. Consequently, new and improved pedagogical approaches and supportive technologies increasingly attract dedicated research and development efforts. Technology-supported lifelong learning is becoming the key to extend formal education not only in primary and secondary, but also in higher educational

learning settings. Specific types of informal learning, such as self-directed learning, become increasingly popular because they enable members of our society to gain knowledge from specialized e-learning repositories, or even from web resources. Especially in these settings, guidance on and information about the learning progress becomes increasingly important, and thus a special focus has to be put on knowledge assessment and feedback methodologies. The creation of test items and the evaluation of tests, however, cannot be performed like in formal settings but requires a creation of test items on the fly [4,6,10].

Our previous work [9,11] and in particular [12] have resulted in a prototype Automatic Question Creator, referred to as AQC hereafter in this paper. The prototype AQC is able to create test items automatically from text corpora [12]. In terms of students' satisfaction of the overall quality of the test items generated, the prototype has demonstrated promising results. However, a deeper investigation highlighted that automatically created *distractors* were perceived by the study participants as being of rather low quality and too easy to detect.

Creating distractors was identified by Haladyna [13] as the most difficult part of the multiple-choice test generation. Furthermore, as our own experience showed, students are exceptionally good at detecting even subtle hints which give away the correct answers. For example,

Q: Which of the following is the term used to describe the task of extracting structured information from unstructured text in natural language?

- A) Natural Language Processing
- B) Information Extraction
- C) Car
- D) Insurance

where B) is the correct answer, C) and D) are easily detectable as wrong answers, whereas A) is a reasonable distractor that may confuse students who do not have a clear understanding of the testing concept *Information Extraction*.

Questionnaires containing unsuitable distractors would therefore only grade the students' ability in pattern detection, but not their knowledge or understanding in the subject area. Consequently, improvements in this subfield of automated question generation will not only allow more students to study more effectively in a self-teaching environment, but also contribute in general to the whole field of automated assessment and lifelong learning.

In order to be able to create good distractors, we have to define what a distractor actually is. The first and foremost property of a distractor adopted by this paper is that: "*a distractor is an unquestionably wrong answer*", as stated by Haladyna [13]. Secondly, these wrong answers are *distractive* in that they should be plausible to students who do not yet understand the course material. Yet however, distractors should be spotted by, and should not confuse, people who already possess the knowledge measured by the multiple-choice test.

In this paper, we developed a novel method for refining distractor generation to ensure that the distractors are plausible yet difficult to detect. The idea of the algorithm reported in this paper is based on the observations by Mitkov

et al. [20], who stated that distractors semantically close to the correct answer were the most plausible distractors, and by Haladyna [13], who elaborated that distractors should resemble the grammatical form, style and length of the correct answer.

In addition, we also make use of the weight measures produced by the existing AQC system to improve the distractor quality. Terms receiving a higher weight are considered to represent more important concepts, and substituting one of these concepts for a different, incorrect one should be able to confuse students who only superficially studied the course material.

Consequently, our implementation of the distractor generation process combines the following approaches:

- Latent Semantic Analysis (LSA) which, according to Landauer et al. [18] is able to capture semantic similarities of words and passages, is used as an estimation for semantic closeness.
- Part-of-Speech (POS) tags which are, as Klein and Simmons [17] stated, a set of grammatical codes. These codes can thus be used to ensure that the answer and the distractors have the same grammatical form.
- Stylometry, which was defined by Holmes [16] as *“the statistical analysis of literary style”*. Our proposed method also take the length differences into consideration.
- The existing AQC weight measures, are used in our distractor refinement to identify important concepts.

Our novel approach extracts concepts from some course material, automatically identifies relevant concepts in the material and uses these concepts to generate multiple-choice questions. The system then applies LSA to the course material, calculates style measures, and uses the results from LSA and the style measures to calculate the similarity between the correct answer and possible distractors. The similarity results together with weights calculated by the AQC are used to rank the distractors, and finally, a filter rejects obviously ungrammatical answers.

The remainder of this paper is structured in the following way: Section 2 introduces the framework on which our research is based and describes related research. Section 3 explains how we create the distractors, and Section 4 shows the setup and results of our distractor evaluation experiment. The paper is summed up in Section 5, and this section also provides ideas for future research.

2 Background and Related Work

2.1 Automatic Question Creator

Research focusing on automated test item creation, as stated by Gütl et al [11], uncovered only a small number of already existing approaches and tools. Furthermore, these tools and approaches were also not flexible and extendable enough to address the practical needs of automated question generation. Consequently, in order to alleviate this situation, we initiated the design and implementation of the AQC.

The architecture of the AQC can be segmented into four parts, the *preprocessing*, the *concept extraction* and the *assessment creation* module, and a *graphical user interface*. The modules work in an independent pipelined fashion, where each module accepts the output of the previous one, and each module can also be controlled and influenced via the user interface.

The preprocessing module accepts unstructured or semi-structured text, performs text cleaning and converts the cleaned result into an XML representation. The resulting XML data is in turn processed by the concept extraction module which performs structural, statistical and semantic analysis, calculates weights for phrases present in the text, and uses these weights to extract suitable concepts. This module is described in more detail in Section 3.2. The final module, assessment creation, determines suitable sentences for each extracted concept, and uses these sentences and the concepts to create open ended, single choice, multiple-choice and completion exercises.

AQC's distractors are created in the module responsible for generating multiple-choice exercises, i.e. the assessment creation module. A brief description of how the system creates distractors can be found in Section 2.4, which also introduces the already implemented generation approaches in AQC.

2.2 Latent Semantic Analysis

The aim of Latent Semantic Analysis (LSA) is, according to Landauer et al. [18], to determine how similar the meaning of words or passages is. This similarity measure can be used to rank distractors, where words or phrases highly correlating with the correct answer are assumed to be close enough to the correct one to confuse students not possessing enough knowledge about the specified test item.

LSA is based on performing mathematical operations on a word-by-context matrix. Context can be a sentence, a paragraph or a whole document, and the original matrix counts how often which words appears in what context. This usually high-dimensional but sparsely populated matrix is then normalized and, via singular value decomposition (SVD), transformed into a product of three different matrices. These three matrices are used to calculate a least-square best fit matrix of the original one. The new matrix is understood to better capture the relationship between words and their contexts than the original one.

2.3 Stylometry

Stylometry is, according to Holmes [15] concerned with extracting measurable patterns describing the style of a text. These patterns support us in the task of ensuring that the style of the incorrect answer sentences matches the one of the correct sentence as closely as possible. The most commonly used features, as identified by Abbasi and Chen [1], are:

- Lexical features, which capture word or character based statistics.
- Syntactic characteristics, providing statistics on function words, punctuation and POS tags.

- Structural properties, which consider text organization and layout.
- Content-specific elements, mainly usage statistics over important keywords and phrases.
- Idiosyncratic features, like misspellings, grammatical errors or incorrect usage of words.

However, as we compare sentences which only differ in a small amount of words, not all of these features can be used. Specifically, punctuation, structural properties, phrase usage and the counting of misspellings are not applicable to our distractor sentence style comparison.

2.4 Existing Distractor Generation Approaches

Past and current distractor generators mainly only consider single-word answers, or they use simple algorithms for creating multi-word distractors. Single word term distractors are commonly selected by word frequency (for instance by Brown et al. [5]), by finding semantically related terms via WordNet (like [11] or [20]) and, in the case of grammar tests, by performing rule-based changes to the word form and tense (see [7]). Multi-word distractor generation commonly tries to obtain related multi-word terms via WordNet. However, as this method rarely produces results, Mitkov et al. [20] select noun-phrases with the same head as the answer, and Gütl et al. [11] split the phrase into n-grams and randomly select the longest related n-grams from WordNet, which frequently results in a distractor phrase with only one changed word.

The approach most closely resembling ours was published by Aldabe and Maritxalar [2]. This approach also uses LSA to find distractors, however, the distractors are single-word only, while we focus on important multi-word concepts. Moreover, in contrast to our work, they use manually created blanks, without employing style measures. Furthermore, their published results are only available for the Basque language.

3 Distractor Generation

3.1 System Composition

Figure 1 illustrates our distractor generation process as a pipeline of three independent modules. The system as a whole accepts free-form text documents as input and generates a list of distractors as its output. The individual modules, which are described below, include the Concept Extraction, the Distractor Selection and the Distractor Refinement module. Communication between the modules is also performed in a pipelined fashion, where each module accepts a list of items from the previous module. This communication approach together with the clear segmentation of the system’s functionality simplifies the replacement of parts or even whole modules and thus allows us to easily extend and enhance this system.

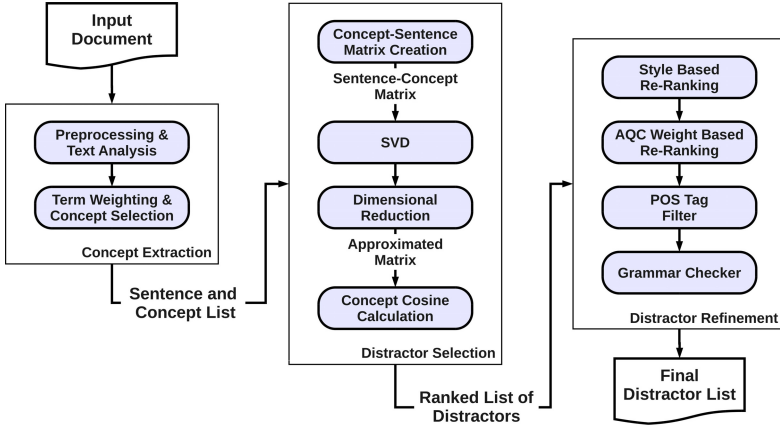


Fig. 1. Overview of the Distractor Generation Process

3.2 Concept Extraction

The concept extraction module, as shown in Figure 1, is actually a part of the AQC, and could be reused without any modification. The most important tasks of this module are the conversion of the input text into a normalized internal representation, the annotation of the text, and the identification of concepts.

The preprocessing and analysis step accepts input texts in various formats, like Microsoft Word, PDF, HTML or plain text, and converts these formats to an internal HTML representation. This internal text is then processed by the General Architecture for Text Engineering (GATE) (see [8]) which performs tasks like tokenisation, named-entity recognition and POS tagging.

Term weights are calculated by performing statistical (like word and term frequencies), semantic (employing WordNet to determine semantic relationships between terms) and structural (for instance if the concept is part of the title, abstract, or chapter heading) analysis of the input text. The numeric values obtained by the analysis are in turn adjusted by user-defined factors and combined into weight measures. Concept selection uses term weights together with weights and annotations provided by XtraK4Me, a module to detect key-phrases in texts (see Schutz [22]), to determine which phrases should be considered to constitute concepts. The most important of these concepts are then used as correct answers for the multiple-choice questions. For a more comprehensive explanation of the details of this module, please see Gütl et al. [11].

3.3 Distractor Selection

After the concept extraction step is finished, lists of sentences and possible concepts are available for this module. The elements of the lists are annotated with POS tags and they contain weight measures provided by AQC. These lists, as can be seen in Figure 1, are used to calculate the semantic similarity between the

correct answer and the distractors and to produce a ranked list of distractors. As stated in the introduction, LSA presents itself as a viable approach to this calculation, and the resulting similarity values can be used to estimate if a term could assume the role of a distractor.

The first task of this module is to generate a matrix which can be used for SVD. The matrix in our case is a concept-sentence matrix, where the rows of the matrix are the possible concepts as created by AQC, and the columns state how often the concept appears in a specific sentence. This matrix is high-dimensional yet sparsely populated. Experiments with our system showed that stemming and stop-word removal improved the final distractor quality, and that adjusting the the values by their row entropy, as proposed by Landauer et al. [18], actually decreased the performance of our system.

The concept-sentence matrix is then transformed via SVD into three component matrices, and the dimensionality of the diagonal matrix is reduced by only keeping the highest values of this matrix. The matrix product of the transformed and the reduced matrices provides an approximation of the original concept-sentence matrix, which, as stated above, is able to capture hidden semantic relations between the different concepts. The SVD transformation is implemented by the Lingpipe [3] toolkit, which is used in our system.

The final step of this module takes the correct answer provided by AQC, and selects its row from the approximated concept-sentence matrix. This row, in turn, is used to calculate the cosine between the correct answer and all possible distractor concepts by using the appropriate rows from the approximated matrix. The resulting distractors are then sorted by their respective cosine values, where higher values indicate higher quality distractors. Though many distance measures for determining the relatedness of two vectors are described in literature (like the Euclidean distance, the maximum norm or even correlation between the values of two vectors), experiments showed that the best results were achieved by using the cosine similarity measure.

3.4 Distractor Refinement

Using the ranked list of distractors produced by the distractor selection module, and the POS and weight annotations provided by the concept extraction we are finally performing the last steps of our distractor generation. These steps, as shown in Figure 1, re-rank the distractors and perform filter operations to discard obviously ungrammatical distractors.

As stated above, the conditions imposed by the fact that the amount of different words between distractors and the correct answer is very small, only a selected amount of stylometric features can be feasibly used. These features are:

- Lexical: the number of characters and digits, character and digit uni- and bi-gram counts, the number of words and the average length of words.
- Syntactic: the POS-tag uni-gram count.
- Content-specific: the word frequency.

The style value differences between the correct answer and possible distractors are converted into numeric values and then used to re-rank the list provided by the distractor selection module. These numeric values are obtained by calculating the cosine for uni- and bi-gram counts, and by calculating the difference and normalizing it by the average of the two values for all other style measures.

The next re-ranking step uses the term-weights created by the AQC, to determine the importance of the extracted concepts. Though we initially assumed that important concepts would provide good distractors, our experiments did not provide a clear correlation between the term-weights and the viability of these terms as a distractor. Though this re-ranking step had only a slight impact on the selection process, we nevertheless kept this step and plan to examine its effects in a future study.

A POS tag filter is responsible for ensuring that the grammatical forms of all possible answers match. This filter extracts the POS tag of the head word of the correct answer concept and compares it with the tags of the head word of all distractor concepts. The current implementation of our system performs exact POS matches and rejects the non-matching concepts, however, conversions between similar word forms, like singular and plural, would be possible, and these methods are scheduled for inclusion in a future version of our system.

The concluding refinement step performs a more sophisticated grammar check than the POS filtering step above. Here all blanks generated by the AQC are filled with the correct answer and with possible distractors, and the resulting sentences are examined. As even sentences with the correct answer sometimes contain grammatical errors or are flagged as being not completely correct, we had to relax our rejection criteria, and only discard distractors which would introduce additional errors. The actual checking is performed by the LanguageTool (see [21]) which is provided as open source and written in Java. Furthermore, this tool contains human-editable grammar rule files which allowed us to fine-tune the evaluation algorithm for our needs.

4 Experiment and Results

4.1 Weight Influence Estimation

Different weight measures obviously influence the re-ranking process to a different degree. To estimate these influences, we trained our system based on data obtained from [11], and consequently used the MIT OpenCourseWare lecture on “Project Management for Construction”, provided by Hendrickson [14], as source material.

The AQC is responsible for generating the multiple-choice questions and the concepts, which were used by our system to calculate a list of all relevant weight measures. The resulting list was expanded by a column containing distractor-fitness values between 0 and 1000 where 1000 indicated a perfect distractor, and 0 indicated that the concept should not be used. Multiple linear regression was then performed with these fitness values as the dependent variable and with all weight measures as independent variables. Examining the regression results

showed that LSA similarity weights had the strongest influence on ranking of the distractor, followed by character-bi-gram differences and POS-tag uni-gram differences.

4.2 Experimental Setting

Evaluation of our distractor generation was performed in the context of the “ISR Natural Language Processing Case Study”, which examined the newest version of the AQC based on an introductory text to Natural Language Processing (NLP). Our aim in this experiment was to show that the re-ranking influences obtained for the “Project Management for Construction” were general enough to work even for non-related topics, and thus we did not specifically adapt our distractor generation approach to this new material.

Participants of the experiment had to study an introduction to NLP written in English. They need to summarise this text, answer some questions concerning this text, to extract important concepts and also had to answer some questions intended to evaluate the knowledge of a student concerning the studied text. As the last task, the participants had to rank three distractors created by AQC and three distractors created by our system based on their assessments of each term’s suitability as a distractor.

The experiment was conducted by nine tutors and PhD students who served as an expert committee to evaluate the quality of the generated distractors. Two of them are female and seven male with an average age of 33. All of the experts are German native speakers with English as a second language.

We would also like to explain the reasons why we prefer a small but experienced expert panel to the more general approach of conducting real tests among large groups of students. Varying abilities in the students will add noise to the evaluation results. In addition, without running other assessments, it is hard to separate students into distinct groups of those with good conceptual understanding and those with superficial and easily confusable knowledge. The small expert group, who all have good understandings in the subject matter, on the other hand, are more suitable in making informed judgement. Moreover, teaching experiences make them more aware of student common mistakes, which in turn allow them to select distractors that are pedagogically valuable. Having said that, we do plan in the future work to conduct evaluation on students.

4.3 Results

The ranking experiment, as can be seen from Figure 2, shows that our new LSA and stylometry approach (LSA-STY) for creating distractors was indeed able to outperform the base system. This figure displays the rank of each distractor together with the percentage of how many of these ranks were filled by distractors created with the LSA-STY system versus how many of these ranks were from distractors of the base AQC system. The best distractor has a rank of 1, and the worst a rank of 6. Consequently, a good distractor generation approach should

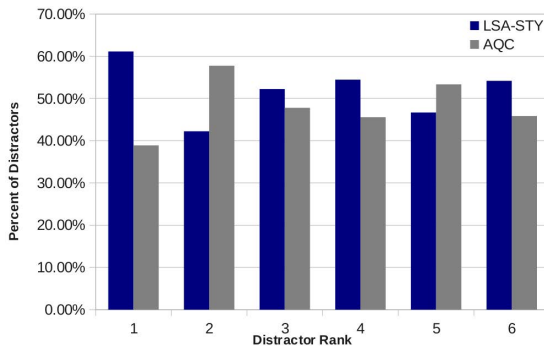


Fig. 2. Result of the distractor ranking. LSA-STY: new system, AQC: base system, lower ranks are better.

posses high percentage values on the low-rank side and low percentages on the high-rank side.

The results for rank 1, the best possible distractor for a question, shows that 61% of these distractors were created by our LSA-STY system versus 39% by the base system. A statistical analysis of this result shows that the two-tailed P-value obtained from the chi-square test equals 0.035, which means that the difference in this distribution can be considered to be statistically significant. Though rank 2 has a high percentage of the second-best distractors generated by the base system (58%), the P-value obtained for this distribution equals 0.14, and thus should not be considered to be statistically significant. The distribution for all other distractors is also not statistically significant.

However, though not significant, the number of the second-best distractors is still high. This indicates, that the WordNet based distractor generation of the AQC will most probably also create viable distractors and should thus not be completely discarded.

5 Conclusion and Future Work

Our system extends the multiple-choice module of the AQC by providing a sophisticated distractor selection method based on the NLP approaches, namely, LSA and stylometry. The distractor selection considers previously extracted concepts, ranks them via LSA, re-ranks them based on stylometry and AQC-provided weight measures, and uses POS tags and a grammar checker to remove unsuitable candidates. Preliminary experiments performed by nine tutors and PhD students showed that our new approach was able to produce the best distractors for a significant number of questions.

Though our initial experiments showed an improvement in distractor quality, a large scale test in a classroom setting has to be performed. This test should follow the approach provided by [19] and [2], who split the answered questions according to the marks awarded into a high- and a low-achieving group, and checked if the

distractors were able to discriminate between these groups. Good question items should confuse low-achieving students more often than high-achieving ones.

Furthermore, due to resource limits, our system was only trained with an excerpt from the “project management in construction” course from MIT OpenCourseWare and tested with an introductory text to Natural Language Processing. The system will be tested with different course material spanning a wide range of subjects. Additionally, when examining these subjects, domain-experts will be employed to manually create some distractors for comparison purposes, and a sub-study should identify how and why these experts select distractors, and might even produce new distractor-weighting techniques.

Another interesting approach would be a combination of the base system and our LSA and stylometry based approach. Though the difference in the number of second-best distractors was not statistically relevant, the WordNet based approach of the base system nonetheless produced a high number of them, and thus also generated viable distractors. This hybrid method could in turn use the distractor-weighting techniques obtained by studying experts to decide which distractor to select.

References

1. Abbasi, A., Chen, H.: Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace. *ACM Trans. Inf. Syst.* 26, 7:1–7:29 (2008)
2. Aldabe, I., Maritxalar, M.: Automatic Distractor Generation for Domain Specific Texts. In: Loftsson, H., Rögnvaldsson, E., Helgadóttir, S. (eds.) *IceTAL 2010*. LNCS, vol. 6233, pp. 27–38. Springer, Heidelberg (2010), http://dx.doi.org/10.1007/978-3-642-14770-8_5
3. Alias-i: Lingpipe 4.0.0 (2010), <http://alias-i.com/lingpipe> (access date: December 16, 2010)
4. Bransford, J.D., Brown, A.L., Cocking, R.R. (eds.): *How People Learn: Brain, Mind, Experience, and School: Expanded Edition*. The National Academies Press (2000)
5. Brown, J.C., Frishkoff, G.A., Eskenazi, M.: Automatic question generation for vocabulary assessment. In: *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, HLT 2005*, pp. 819–826. Association for Computational Linguistics, Morristown (2005), <http://dx.doi.org/10.3115/1220575.1220678>
6. Chang, V., Gütl, C.: Generation y learning in the 21st century: Integration of virtual worlds and cloud computing services. In: Abas, Z.W., Jung, I., Luca, J. (eds.) *Proceedings of Global Learn Asia Pacific 2010*, pp. 1888–1897. AACE, Penang (2010)
7. Chen, C.Y., Liou, H.C., Chang, J.S.: Fast: an automatic generation system for grammar tests. In: *Proceedings of the COLING/ACL on Interactive Presentation Sessions*, pp. 1–4. Association for Computational Linguistics, Morristown (2006), <http://dx.doi.org/10.3115/1225403.1225404>
8. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: A framework and graphical development environment for robust NLP tools and applications. In: *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics* (2002)

9. Gütl, C.: Automatic limited-choice and completion test creation, assessment and feedback in modern learning processes. In: LRN Conference 2008, Guatemala (February 2008)
10. Gütl, C., Chang, V.: Ecosystem-based theoretical models for learning in environments of the 21st century. *International Journal of Emerging Technologies in Learning (iJET)* 3(1) (2008)
11. Gütl, C., Lankmayr, K., Weinhofer, J.: Enhanced approach of automatic creation of test items to foster modern learning setting. In: *Proc. of the 9th European Conference on e-Learning*, pp. 225–234 (November 2010)
12. Gütl, C., Lankmayr, K., Weinhofer, J., Höfler, M.: Enhanced approach of automatic test item creation to foster modern e-education. *Electronic Journal of e-Learning* (2011) (to appear)
13. Haladyna, T.M.: *Developing and Validating Multiple-Choice Test Items*, 3rd edn. Lawrence Erlbaum Associates (2004)
14. Hendrikson, C.: *Project management for construction* (2000), <http://pmbook.ce.cmu.edu/index.html> (access date: November 17, 2010)
15. Holmes, D.: Authorship attribution. *Computers and the Humanities* 28, 87–106 (1994), <http://dx.doi.org/10.1007/BF01830689>
16. Holmes, D.: The Evolution of Stylometry in Humanities Scholarship. *Literary and Linguistic Computing* 13(3), 111–117 (1998), <http://llc.oxfordjournals.org/content/13/3/111.abstract>
17. Klein, S., Simmons, R.F.: A computational approach to grammatical coding of english words. *J. ACM* 10, 334–347 (1963), <http://doi.acm.org/10.1145/321172.321180>
18. Landauer, T.K., Foltz, P.W., Laham, L.: An introduction to latent semantic analysis. *Discourse Processes* 25, 259–284 (1998)
19. Mitkov, R., Ha, L.A.: Computer-aided generation of multiple-choice tests. In: *Proceedings of the HLT-NAACL 03 Workshop on Building Educational Applications Using Natural Language Processing*, vol. 2, pp. 17–22. Association for Computational Linguistics, Morristown (2003), <http://dx.doi.org/10.3115/1118894.1118897>
20. Mitkov, R., Ha, L.A., Varga, A., Rello, L.: Semantic similarity of distractors in multiple-choice tests: extrinsic evaluation. In: *GEMS 2009: Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*, pp. 49–56. Association for Computational Linguistics, Morristown (2009)
21. Naber, D.: *A Rule-Based Style and Grammar Checker*. Diploma thesis, University of Bielefeld (2003), <http://www.language-tool.org/> (access date: January 2011)
22. Schutz, A.: *Xtrak4me - extraction of keyphrases for metadata creation*. SmILE: Semantic Information Systems and Language Engineering Group (2008), <http://smile.deri.ie/projects/keyphrase-extraction> (access date: November 17, 2010)