Review of the Main Approaches to Automated Email Answering

Eriks Sneiders[1]

[1] Department of Computer and Systems Sciences, Stockholm University
Postbox 7003, SE-164 07, Kista, Sweden
eriks@dsv.su.se

Abstract. There were 108.7 billion business emails sent daily in 2014, many of them to contact centers. A number of automated email answering techniques have been explored in order to ease the burden of manual handling of the messages. Most techniques stem from three text retrieval approaches – text categorization by machine learning, statistical text similarity calculation, matching of text patterns and templates. The paper discusses the previous research in automated email answering and compares the techniques.

Keywords: Automated email answering, message answering, email categorization, email classification.

1 Introduction

Despite losing its popularity to social media, email remains the most pervasive form of communication in the business world with 108.7 billion daily messages in 2014 [1]. For companies and governmental organizations, email and web-based messaging services are a vital communication channel with customers and citizens, along with telephone. And is this channel cheap? A contact center in Sweden may have a cost as much as 12 EUR per message [2].

Automated email answering is not a common practice in business settings. One reason why companies may be reluctant to send automatically generated answers is fear of losing contact with their customers, as well as fear of lost sales opportunities, according to our private communication with a few contact centers in Sweden. On the other hand, a system that retrieves draft answers for contact center agents is a welcomed alternative. Automated text message answering has a future also without the traditional inbox. For example, an interactive system that handles problem reports (e.g., “trouble tickets”) can benefit from an immediate feedback or solution. A similar feature can be useful in community question answering (web forums).

Tang et al. [3] offer a comprehensive overview of the main email tasks – spam detection, email categorization, contact analysis, email network property analysis, email visualization – and the corresponding techniques. This paper discusses the previous research in automated email answering, which is not as popular as automated question answering [4], yet still has a number of publications, many dated by the pre-social-media era. The method of obtaining the publications was our own years-long research and Google Scholar. We submitted a number of queries to Google Scholar

Á. Rocha et al. (eds.), *New Advances in Information Systems and Technologies*,
Advances in Intelligent Systems and Computing 444,
DOI 10.1007/978-3-319-31232-3_13

and explored the top 100 links. Also, we checked the reference section of the selected publications. A significant criterion for selecting a publication for the review was presence of an experimental evaluation of the technique.

2 Machine Learning for Email Categorization

If all messages in one text category have the same standard answer, automated email answering becomes a text categorization task.

Busemann et al. [5] tested a number of machine learning techniques – k-NN, Naïve Bayes, RIPPER, SVM. The data was 4490 email messages, sent to a technical support helpdesk, in 47 categories with at least 30 messages in each category. Optional input for machine learning was presence or absence of linguistic constructions frequent in questions and problem descriptions: negation at the sentence or phrase level, yes/no and who-when-what-why-where-which-questions, a declarative immediately preceding the question. All experiments were carried out using ten-fold cross validation. The best performing system was SVM, with accuracy (the share of correctly suggested standard answers) of 56% for a single answer, and for 78% of the email messages the correct answer was among the top 5 suggested answers.

Tobias Scheffer [6] explored SVM and Naïve Bayes for email answering in the settings of Internet-mediated education. Roughly 72% of all incoming messages could be answered by 9 standard answers; the most popular inquiry alone covered 42%. The performance of text categorization was assessed by receiver operating characteristic (ROC) curves that showed the relationship between true positive and false positive category placements as the discriminating threshold varied. A large area under the curve shows a large probability of true positive over false positive category placements. 7-to-20-fold cross validations were performed and their results averaged. SVM demonstrated superior performance, 80-95% area under the curve, even with as few as seven labeled positive training examples. SVM significantly outperformed Naïve Bayes in 8 of 9 text categories.

The same team did a parallel study [7] with email of a large online shop. A set of 805 messages was manually divided into 19 partitions with at least 10 messages in a partition where all messages in the partition had semantically equivalent answers. 10-fold cross validation with SVM reached categorization accuracy (precision equals recall) 42%. One problem was that different partitions contained similar query messages while their answers were different; determining the difference required additional information from the shop's order database. After merging the similar partitions, a number of precision-recall measurements showed 10-20% precision improvement over the same recall values.

Mercure was a 4 years long study with the goal to explore the automation opportunities in processing email sent to a contact center [8]. Two approaches were investigated: one did message categorization with machine learning, the other one pursued the paradigm of text retrieval. Text categorization was tested by k-NN, Naïve Bayes, and RIPPER . The data was 1568 inquiry-reply pairs in the domain of investor relations at a telecommunications company. Categorization accuracy was about 50%. The main cause of errors was the fact that some messages dealt with more than one subject or were a part of a multi-message conversation. Similar tests with single-subject emails yielded 90% accuracy for 5 categories, 80% for 10 categories, and 67% accuracy for 22 categories.

Yang and Kwok [9] compared the K-means++, k-NN, and Naïve Bayes algorithms. The data was 3015 emails in 200 text categories, each having 5-13 messages, in the domain of computer technical support. 10 experiments were conducted, each time 1000 training messages used. In average, K-means++ correctly classified 96.2% of the messages, Naïve Bayes – 89.5%, k-NN – 75.5%.

Hewlett and Freed [10] helped employees of a contact center compose email replies. Given the query email, the system retrieved the answers of the 8 most similar archived messages, using cosine similarity as the text similarity measure. The archived messages had their term vectors stored and readily available. The employee selected an appropriate answer, thus creating an opportunity for machine learning: the message of the selected archived answer is most similar to the query email, the messages higher in the top-8 list are not. The system used a version of the Margin Infused Relaxed Algorithm to update the stored term vectors of the selected message and the messages above it in the top-8 list.

The performance evaluation data was email sent to the Hewlett Packard helpdesk. 8604 archived message-answer pairs were selected. Of them, 475 were grouped into 36 text categories, where all messages in one category had identical bag-of-words answers. The system matched each of the 475 messages to the 8604 messages using cosine similarity. In 231 cases of 475 a relevant answer appeared in the top-8 list, which is 49% of the cases. After adding the machine learning component to the test, this share increased to 60%.

Matching an email message to a list of FAQs on a website is another way of answering. Itakura et al. [11] uses SVM in a less traditional way. The central concept is a feature vector with similarities between one message and the FAQs. Let's say there are *n* FAQs in a list. An *n*-dimensional feature vector contains *n* modified Jaccard coefficients that quantify the similarity between each FAQ and the message. The modified Jaccard coefficient is calculated by adding weight to domain keywords considering their inverted document frequency. SVM is then trained with such feature vectors, one vector per message, not document vectors based on words in the message or FAQ texts.

The performance evaluation used 4 FAQs and 1845 email messages, 545 of which corresponded to the FAQs. The language was Japanese. 22 domain keywords and their synonyms were used in calculating the modified Jaccard coefficient. The resulting precision/recall figures were roughly 98/90, 85/70, 95/85, and 60/95 percent for the 4 text categories defined by the 4 FAQs.

The previous email categorization tasks (not automated answering) have used more features for machine learning than just message text. Alberts and Forest [12] have tested two sets of features – lexical features, i.e. message text, and non-lexical features, such as use of bold and capital letters, number of recipients of the message, presence of a mailing list, the sender being in the same social network as the receiver, presence of "RE:" and "FW:" in the subject line, message signed by an official signature or only by the first name, etc. A test with k-NN and 1700 messages in French from two workplace inboxes showed that categorization accuracy with non-lexical features is considerably lower than that with lexical features. The most discriminative non-lexical features were: the sender in the same social network as the recipient, message to or from a mailing list, the number of carbon-copy recipients, use of bold letters, bilingual message.

3 Statistical Text Similarity without Machine Learning

Locating one or several inquiries in a messages and selecting one or several standard answers is a more challenging task than document categorization. Malik et al. [13] work with sentence matching. The system maintains a large number of standard answers and a variety of tag-questions, like FAQ questions, attached to each standard answer. When a query email arrives, the system matches sentences in the query to the tag-questions. When the system matches two sentences, it considers the distance between concepts in the sentences, which is obtained from WordNet. If one of the two words does not exist in the dictionary, then edit distance between the words is calculated.

During the preparation phase, a "training" system examined a large number of archived email message-reply pairs and assigned the tag-questions to the standard answers. To spot a piece of a standard answer in the reply text was easy; more difficult was to map a question in the inquiry message to that piece of a standard answer. In order to identify a question and the answer in a message-reply pair, the system had a list of domain-specific uni-, bi-, and trigrams; questions and answers had to contain these domain-specific n-grams. Further, the system calculated word overlap between the question and the answer, adjusted with respect to inverted document frequency of different words.

The experiment data was 1320 message-reply pairs and 570 standard answers in the domain of mobile phone services. On average, there were about 2 questions per customer inquiry. The email answering system was "trained" by 920 message-reply pairs: the tag-questions were assigned to the standard answers. Tested with 400 message-reply pairs, the system generated the same reply as the humans did in 61% of all cases. In 73.4% of the cases the system generated partially correct replies.

Mercure [8], whose machine learning and text categorization module was introduced in the previous section, tested retrieval of archived message-answer pairs in order to reuse old answers. The system compared query messages with archived email messages applying cosine similarity; term weights were made of term frequency and inverted document frequency. Precision of the retrieval reached 57.9%. A difficulty of email text matching is the lexical gap - a large variety of wordings used by different people to express the same thing. The vocabulary of the answers written by the employees of the contact center, on the contrary, is more uniform. It would be good if instead of comparing inquiry messages (dispersed vocabulary) we could compare their answers (uniform vocabulary). The idea was implemented through query expansion. During the preparation phase, the system measured co-occurrence of words between archived messages and their answers. When a query message arrived, the system used this co-occurrence to find "synonyms" for the query expansion. The precision improved to 62%. In one text category, where the reply was a generic redirection to a web page, the improvement was from 51% to 80.1%.

Alfalahi et al. [14] reused the Mercure's idea and introduced the concept of shadow answer. A shadow answer is a query message re-written in the terms of its likely answer. The term translation is done according to message-answer term co-occurrence between archived messages and their answers. The shadow answer becomes a new query in the database of previous answers. A test with archived emails as queries was conducted; the researchers were looking for the position of the original answer in the result list. Messages were bags-of-words – unigrams and/or bigrams. The best performance was achieved with bigrams only, the result was mediocre: the average position of the original answer of an archived email submitted as a query was 28.

While most text categorization applications in email answering assume that each message in one text category has the same standard answer, for Weng and Liu [15] message categorization is only a half way to the answer. Deployment of the system begins with selecting at least 10 domain concepts attached to a number of text categories; one concept may be attached to several categories. The concepts contain weighted domain terms, where the weights are calculated considering "inverted concept frequency" (a term may appear in several concepts) and term frequency in the training data. When a query message arrives, its term weights are multiplied with the concept term weights, and the "heaviest" concepts lead to a category placement. The category placement and the "heaviest" concepts together determine one or two standard answers.

The evaluation used 612 FAQs and their answers in the domain of Windows NT/2000. 191 simulated user emails and downloaded forum posts were used as queries. Precision and recall were around 80%, with the recall being slightly lower than the precision.

4 Answer Generation

Answer generation is a minority approach in automated email answering. It is not a technique. Rather, it is the outcome of the application of the techniques. Still, the outcome is so different from the usual standard answer selection that it deserves a separate section. We have identified three studies. Two studies involve information extraction and filling answer templates; the third study is answer generation by collating sentences from previous answers.

Kosseim et al. [16] use information extraction templates in order to (i) identify the query message – the purpose, the sender, etc., (ii) extract names entities from the query message, (iii) extract relations between the main concepts, and (iv) capture domain-specific relations. The next step is semantic validation – the system verifies whether the extracted data and the respective templates all together make any sense as an answer. The third step is analysis of the obtained information and querying some external sources for new data to complete the answer. Finally, the system fills the answer template with the data and generates the answer text.

The prototype was tested with 191 email messages about printer-related problems. 122 messages were used for analysis, 69 messages for the test. 27.7% of the test messages were answered correctly, 13.3% were answered incorrectly. 38.7% of the test messages were correctly redirected for manual processing, 20.3% were redirected incorrectly. In total, 66.4% of the responses were correct.

Probability that a word appears in a certain context is central for Al-Alwani [17]. The system answers emails that belong to a few text categories; each text category has one standard answer template that needs to be filled with details before the answer is sent to the user. The system has two dictionaries that are populated during the training phase. One dictionary contains words and the probability that a word appears in a message of a certain text category; it is used for message classification. The second dictionary is used for information extraction from query messages. It contains words and the probability that a word appears next to an item to be extracted from a query message, e.g., a product name, a meeting place and date.

Pre-processing of the messages starts with part-of-speech tagging and lemmatization. Negations are merged with their target words. For example, "didn't receive" becomes "not-receive", a new terms different from "receive". Considering negated words as new terms is not a common practice in automated email answering. After that prepositions, pronouns, interjections, and conjunctions are removed. The message becomes a bag-of-words. Synonymy of the remaining words is resolved. The query message is categorized using the first dictionary. When the system knows which answer template is to be filled and what items need to be extracted from the query message, the second dictionary is applied in order to locate the items in the query text. There is no external source of additional information for the answer. Information extraction and filling the answer template serves only the purpose of a more personalized answer.

Three text categories with 400, 200, and 400 messages were selected for the performance evaluation. Dividing the data into the training and test collections is not discussed. The precision/recall values were 80/69, 72/60, and 76/65 percent.

Yuval Marom and Ingrid Zukerman have worked with email answer generation where the answers are collated by reusing sentences from the previous human-written answers to similar messages. One of their earlier works [18] explains the foundation of the approach. Answers to recurring email inquiries to a contact center consist of two parts – generic information common for a number of similar inquiries and details specific to the particular inquiry. Marom and Zukerman applied multi-document summarization techniques in order to create the generic portion of the email answer.

The process is following. Similar archived responses are clustered so that one model response can be generated. The model response is built by collating the most representative sentences from the cluster responses. The quality of the model response depends on the semantic compactness of the cluster. The more similar the cluster responses are, the more accurate and representative the model response is.

The domain of the email corpus was helpdesk at Hewlett Packard. The corpus was 8000 message-response pairs clustered into topic-related datasets with 300-1500 pairs in each dataset. The quality of a model response was assessed by comparing it with each original response in the cluster. Precision and recall were measured for each comparison, then the average was calculated. Precision gave the proportion of the words in the model response that matched those in the original response; recall gave the proportion of the words in the actual response that were included in the model response. Generally, the achieved precision was above 50%, recall below 50%.

5 Text-Pattern Matching for FAQ Retrieval

Text-pattern matching is another minority approach. Sneiders [19] has developed a technique that operates a set of manually crafted text patterns assigned to FAQs. A text pattern resembles a regular expression. It contains stems of words and their synonyms. It can match phrases and stand-alone words, also compound words. Each FAQ has one or several required text patterns (they have to match a query) and one or several forbidden text patterns (they must not match the query). Experiments in two languages (Swedish and Latvian) and two domains (insurance and telecom) showed consistent results: if the system did retrieve an answer, the answer was correct in about 90% of the cases. The recall values were 68% and 76% in the respective language and domain.

A test of the same technique with almost 10 thousand emails in the domain of Swedish social welfare [20] showed comparable results. Five FAQs formed five text categories with 2517 messages. Email categorization with text-pattern matching reached precision around 90% and recall values 59, 65, 76, 44, and 59 percent per category. The baseline SVM and single-term bag-of-words reached precision/recall values 69/69, 69/86, 73/89, 70/58, and 63/80 for the same five text categories.

6 Task Related Email Categorization

Sneiders at al. [20] argue that a good email answering technique has to identify the context of the inquiry and the request stated in the inquiry; topical text similarity is not enough. The request designates the purpose of the inquiry. Although resolving the purpose of an email message is not automated email answering, it is a closely related research area.

There exists research that applies speech-act theory in order to categorize workplace email messages according to the purpose of the message, not the topic. Khosravi and Wilks [21] analyzed 1000 email messages and developed a text-pattern matching system that tagged sentences in messages with 10 nuanced request labels, where requests for action, information, and permission were expressed directly or implicitly in statements and questions. Corston-Oliver et al. [22] had an ambition to create a system that would analyze an email message and add action items to the receiver's to-do-list. Sentences in a message were labeled "salutation", "social chit-chat", "task", "proposal to meet", "promise", "farewell", and various components of email signature. Best candidates for action items were "task", "proposal to meet", and "promise".

Cohen et al. [23] tagged entire messages, not individual sentences, according to the intent of the sender. The intent was identified with help of a small ontology of email acts, where the main actions were "request", "propose", "amend", "commit", "deliver", and the subjects of these actions were "information", "meeting", "data", etc. Goldstein and Sabin [24] took a broader look at email tasks and defined 12 email genres by task, including not only the familiar directives, commitments, requests for information, but also expression of feelings, document forwarding, ads and spam, etc.

Lampert et al. [25] added details to the analysis of requests and commitments in workplace email. Requests and commitments may be conditional or unconditional, explicit or implicit. An example of implicit request is “Can you send me the curves and trades for Jan 18?” Although the request appears as a yes/no question, the receiver is expected to act upon it, not to answer it. Neither do we answer rhetorical questions or pleasantries – polite social utterances like “How are you?” Requests and commitments may be made on behalf of the writer or a third person. Clarifying these details proved crucial for improving agreement between human annotators who labeled utterances containing requests and commitments. Lampert et al. [26] pinpoints the difficulties in identifying requests and commitments, where the most prominent one is locus ambiguity: while human annotators tend to agree that the message contains a request or commitment, they may not agree on exactly which utterances contain them.

7 Highlights of the Approaches

We have selected a few features of email answering, features that we find important, in order to highlight the strengths and weaknesses of the automated email answering approaches. Table 1 shows the features and how well they are served by each automated email answering approach. Answer generation is included as the feature “correctness of custom answers”.

Table 1. Features of the email answering approaches.

	Machine learning text categorization	Statistical text similarity	Text patterns, templates
Correctness of fixed answers	Medium	Medium	Good
Correctness of custom answers	N/A	Medium	Medium
Database data in the answer	No	No	Yes
Nuances in the query text	Poor	Poor	Good
Diversity of answers	Large	Unlimited	Small
Domain dependency	Medium	Little	Significant
Same domain, new language	Easy to medium	Easy	Difficult

When people send an email they expect first of all a *correct answer*. Unfortunately, most email answering techniques, except text-pattern matching, reach the level of answer correctness suitable mostly for internal use by contact center agents.

Ability to deliver *custom answers* is limited. Only a few systems have been developed, their performance is mediocre, or the tests are not convincing.

Inclusion of *database data* into the answers is very limited – only Kosseim et al. [16] have customized the answers by data from a knowledge base. The line of development by Sneiders [27]→[19] may have a potential: templates for question answering from a structured database could be applied in email answering, but it has not been tested.

The bag-of-words document representation, which is typical for statistical text similarity calculations with and without machine learning, has a limited ability to *distinguish nuances in the query text*. Text patterns are better for matching nuanced pieces of text.

With a sufficient amount of training data we can train the system to categorize email into any number of text categories linked to a large *variety of answers*. The difficulty is manual labelling of the training data, unless this data is readily available. Statistical text similarity can be calculated for any two pieces of text. Text patterns and templates can cope with only a small number of answers, therefore they are best used for retrieval of the most popular FAQs.

Statistical text similarity calculations with and without machine learning have limited *domain dependency*, unless domain specific ontological and linguistic knowledge is developed to enhance the system's performance. For machine learning, new training data has to be labeled. As for text patterns and templates – they have to be developed from scratch.

This applies also to *language dependency*.

Most researchers in automated email answering calculate topic-related text similarity based on presence of individual domain specific words in both texts. This most often yields mediocre precision. Sneiders et al. [20] argue that good automated email answering requires a combination of *topic- and task-related* email categorizations (see Section 6) which considers also the purpose of the message.

References

1. Radicati, S.: Email Statistics Report, 2014-2018. Radicati Group, April (2014)
2. Suneson, B.: E-post är dyrare men ger kunden bättre svar. Svenska Dagbladet (April 4, 2005). In Swedish.
3. Tang, G., Pei, J., Luk, W. S.: Email mining: tasks, common techniques, and tools. Knowledge and Information Systems, 41(1), pp. 1-31 (2014)
4. Andrenucci, A., Sneiders, E.: Automated Question Answering: Review of the Main Approaches. In: Proc. 3rd International Conference on Information Technology and Applications (ICITA). IEEE, Vol. 2, pp.514-519 (2005)
5. Busemann, S., Schmeier, S., Arens, R. G.: Message classification in the call center. In: Proc. Sixth Conference on Applied Natural Language Processing, pp. 158-165. ACL (2000)
6. Scheffer, T.: Email answering assistance by semi-supervised text classification. Intelligent Data Analysis, vol. 8, no. 5, pp. 481-493. IOS Press (2004)
7. Bickel, S., Scheffer, T.: Learning from message pairs for automatic email answering. Machine Learning: ECML 2004. Springer Berlin Heidelberg, pp. 87-98 (2004)
8. Lapalme, G., Kosseim, L.: Mercure: Towards an automatic e-mail follow-up system. IEEE Computational Intelligence Bulletin, vol. 2, no. 1, pp. 14-18 (2003)
9. Yang, W., Kwok, L. Improving the automatic email responding system for computer manufacturers via machine learning. In: Proc. Information Management, Innovation Management and Industrial Engineering (ICIII), Vol. 3. IEEE (2012)
10. Hewlett, W.R., Freed, M.: An Email Assistant that Learns to Suggest Reusable Replies. In: Papers from the 2008 AAAI Workshop, Technical Report WS-08-04, pp. 28-35 (2008)
11. Itakura, K., Kenmotsu, M., Oka, H., Akiyoshi, M.: An Identification Method of Inquiry E-mails to the Matching FAQ for Automatic Question Answering. Distributed Computing and Artificial Intelligence, pp. 213-219. Springer Berlin Heidelberg (2010)

12. Alberts, I., Forest, D.: Email pragmatics and automatic classification: A study in the organizational context. Journal of the American Society for Information Science and Technology 63, no. 5, pp. 904-922 (2012)
13. Malik, R., Subramaniam, V., Kaushik, S.: Automatically Selecting Answer Templates to Respond to Customer Emails. In: Proc. IJCAI, Vol. 7, pp. 1659-1664 (2007)
14. Alfalahi, A., Eriksson, G., Sneiders, E.: Shadow Answers as an Intermediary in Email Answer Retrieval. In: Experimental IR Meets Multilinguality, Multimodality, and Interaction, pp. 209-214. Springer International Publishing (2015)
15. Weng, S.S., Liu, C.K.: Using text classification and multiple concepts to answer e-mails. Expert Systems with applications 26, no. 4, pp. 529-543 (2004)
16. Kosseim, L., Beauregard, S., Lapalme, G.: Using information extraction and natural language generation to answer e-mail. Data & Knowledge Engineering, vol. 38, pp. 85-100 (2001)
17. Al-Alwani, A.: Improving Email Response in an Email Management System Using Natural Language Processing Based Probabilistic Methods. Journal of Computer Science 11, no. 1, pp. 109-119 (2015)
18. Marom, Y., Zukerman, I.: Towards a Framework for Collating Help-desk Responses from Multiple Documents. In: Proc. IJCAI Workshop on Knowledge and Reasoning for Answering Questions, Edinburgh, Scotland, pp. 32–39 (2005)
19. Sneiders, E.: Automated Email Answering by Text Pattern Matching. In: H. Loftsson, E. Rögnvaldsson, S. Helgadóttir (eds.): Proc. 7th International Conference on Natural Language Processing (IceTAL), LNAI 6233, pp. 381-392. Springer, Heidelberg (2010)
20. Sneiders, E., Sjöbergh, J., Alfalahi, A.: Email Answering by Matching Question and Context-Specific Text Patterns: Performance and Error Analysis. In: Proc. WorldCist'16, Recife, Brazil, March 22-24. Springer International Publishing (2016)
21. Khosravi, H., Wilks, Y.: Routing email automatically by purpose not topic. Natural Language Engineering, 5, pp. 237-250 (1999)
22. Corston-Oliver, S., Ringger, E., Gamon, M., Campbell, R.: Task-focused summarization of email. In: Proc. ACL-04 Workshop: Text Summarization Branches Out, pp. 43-50 (2004)
23. Cohen, W.W., Carvalho, V.R., Mitchell, T.M.: Learning to Classify Email into "Speech Acts". In: Proc. EMNLP, pp. 309-316 (2004)
24. Goldstein, J., Sabin, R.E.: Using speech acts to categorize email and identify email genres. In: Proc. 39th Annual Hawaii International Conference on System Sciences, vol. 3. IEEE (2006)
25. Lampert, A., Dale, R., Paris, C.: The nature of requests and commitments in email messages. In: Proc. of the AAAI Workshop on Enhanced Messaging, pp. 42-47 (2008)
26. Lampert, A., Dale, R., Paris, C.: Requests and Commitments in Email are More Complex Than You Think: Eight Reasons to be Cautious. In: Proc. Australasian Language Technology Association Workshop, vol. 6, pp. 64-72 (2008)
27. Sneiders, E.: Automated Question Answering Using Question Templates that Cover the Conceptual Model of the Database. In: Natural Language Processing and Information Systems. Proc. NLDB'2002. Springer LNCS series 2553, pp. 235-239 (2002)