Technische Universität Dresden
Medienzentrum

Prof. Dr. Thomas Köhler
Jun.-Prof. Dr. Nina Kahnwald
(Hrsg.)

GeNeMe '13

Gemeinschaften in Neuen Medien

an der
Technischen Universität Dresden
mit Unterstützung der

BPS Bildungsportal Sachsen GmbH
Campus M21
Communardo Software GmbH
Dresden International University
eScience – Forschungsnetzwerk Sachsen
Gesellschaft der Freunde und Förderer der TU Dresden e.V.
Gesellschaft für Informatik e.V.
Gesellschaft für Medien in der Wissenschaft e.V.
IBM Deutschland
itsax – pludoni GmbH
Kontext E GmbH
Learnical GbR
Medienzentrum, TU Dresden
ObjectFab GmbH
Transinsight GmbH
T-Systems Multimedia Solutions GmbH
Universität Siegen

am 07. und 08. Oktober 2013 in Dresden

www.geneme.de
info@geneme.de

C.2 Topic-Based Aggregation of Questions in Social Media

Klemens Muthmann
Technische Universität Dresden, Institut für Systemarchitektur

1 Introduction

Software produced by big companies such as SAP is often feature rich, very expensive and thus only affordable by other big companies. It usually takes months and special trained consultants to install and manage such software. However as vendors move to other market segments, featuring smaller companies, different requirements arise. It is not possible for medium or small sized companies to spend as much money for business software solutions as big companies do. They especially cannot afford to hire expensive consultants. It is on the other hand not economic for the vendor to provide the personnel free of charge. One solution to this dilemma is bundling all customer support cases on special Web platforms, such as customer support forums. SAP for example has the SAP Community Network[1]. This has the additional benefit that customers may help each other.

However the content available on such a site grows rapidly and soon it gets hard for moderators and users to keep an overview of the problems discussed on the page. It gets especially hard to find out if a problem has already been discussed and in many cases it is easier to just open a new discussion, instead of searching if there already is a solution. That way, different threads for the same question exist and in the worst case the true solution is divided over two or more threads. The problem is even larger on a Web wide scale, where multiple forums exist for the same topic area and each might contain its own thread on a similar topic.

Finding such near-duplicate questions is an application of text topic detection. Current algorithms for topic detection however mostly ignore semantic similarities for simpler syntactic ways of matching similar question. Syntax, for the purpose of this paper, means simple matching of the same word occurring in two near duplicate question candidates, possibly modified by some scoring of the words importance for the question. Semantic similarities in contrast are relations between words as well as semantic connections, such as synonymy and hypernymy. In this paper we assume that such relations are important to find near-duplicates of questions with the same content phrased by different people, who might not even be aware of each other. For this reason, this paper presents a semantic topic detection approach on Web forum questions, which considers semantic relations using the generic knowledge

[1] http://scn.sap.com/

base provided by WordNet [10]. The proposed approach finds questions with near-duplicate content to a query question. That way, equal questions are easy to group and to show to users with a similar question, in a condensed way.

The contribution of this paper is the examination of shallow semantic topic detection on question posts from Web forums, using questions from different forums. It also includes a detailed discussion of the possible results. In addition we provide a comparison to naïve implementations and a state of the art approach.

2 Related Work

The area of Topic Detection was established with the *Topic Detection and Tracking* (TDT) conferences by Allan et al. [1]. Topic Detection finds topic groups in a collection of news stories and assigns them to the correct group. A story is a clearly marked piece of text as short as a single paragraph, a forum post or even as long as a whole Web page. It is defined as *A Seminal event or activity along with all directly related events and activities*. Allan et al. [1] describe the research problems of Story Segmentation, First Story Detection, Cluster Detection, Tracking and Story Link Detection. Cluster Detection and Story Link Detection are important for this paper. However the definition of a topic does not fit, since it requires an event or activity. The publications of TDT propose probabilistic and vector space based approaches for topic detection on news stories from the TDT2 corpus. The probabilistic approach [11][25][27] models each topic as a language model such as a Hidden Markov Model, which was proposed by Leak et al. [11] or simple unigram models as developed by Yang et al. and Yamron et al. [25][27]. These approaches generalize well to all texts and are even language independent. On news stories the approach proposed by Leak et al. achieves only mediocre results with Precision of 53% and Recall of 67% for the cluster detection task. This however is still better than the results by Yang et al. [27] and Yamron et al. [25].

In contrast to the probabilistic approach, the vector space approach models topics and news stories as vector from the vector space of all words [27][6][7][12][2][17][4].

Both variations were only tested on news topics using a clean and limited vocabulary.

Online discussions in particular are examined for example by Tang [18]. She builds graph structures through comparison of matching keywords in online discussions. Using these keyword graphs her approach is able to cluster the blogosphere, online discussion boards or similar Web 2.0 applications into topics. However the detected topics only build an overview of hot topics and are not associated to groups of similar questions. Bengel et al. [3] uses a set of predefined concepts and a keyword index for each concept to categorize chat messages into hot topics similar to Tang. Even though such messages show similar properties as forum posts, such short utterances are not the focus of this work.

Work on Semantic Topic Detection was introduced for example by Wang et al. [20]. They extract terms from research papers, considering synonyms and hypernyms, using data collected from WordNet. This approach shows that a semantic topic detection approach is able to achieve high quality results on research papers. The approach presented in this paper will build on some of these ideas but add word co-occurrences and show performance on Web forums.

Topic Detection in Web forums was examined by [22][24][26]. However, these algorithms usually consider forum pages or whole threads – not question posts – as first class entities.

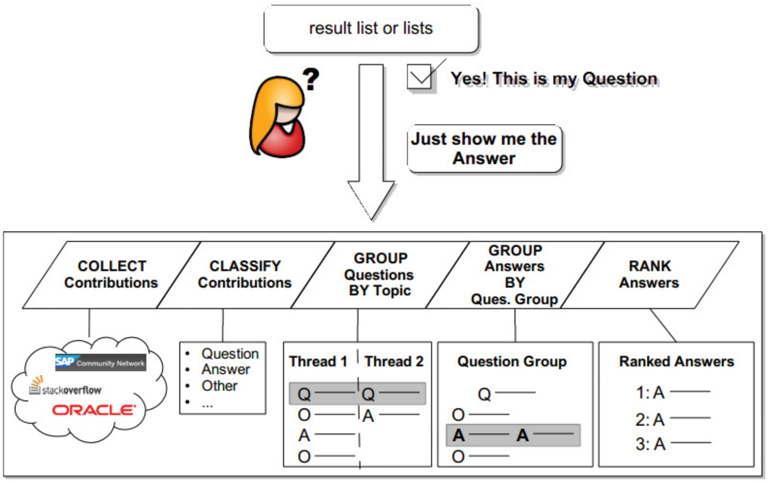

Figure 1: The complete Effingo process to answer questions based on content from social media sites.

3 The Effingo Vision

The problem discussed here is part of a complete system for question answering, using content from social media sites, mainly Web forums. The system is called Effingo and follows the approach presented in Figure 1. The scenario is a user who wants to solve a question, using the World Wide Web. For this purpose she has to manually search through result lists, provided by search engines. Even if the question was found, it is still possible that there is no answer at that location or that the answer is not complete. In such a case the user would actually like to mark the question and click a button to tell the system to search for the complete answer, which might be provided from the content of other discussions with near-duplicate questions. To provide such functionality the Effingo system proposes a five step process. The first step crawls the Web, collects pages from forums and extracts the content to some unified data schema such as proposed by Pretzsch et al. [16]. Then questions, answers and other posts are separated, as shown in [5][9]. The third step arranges questions by topic as proposed in this paper. Since forum topics do not necessarily relate to an event or a seminal activity, the following definition replaces the one proposed by Allan et al. [1]: *Two question posts are topical near-duplicates if it is possible to formulate an answer satisfying both at least partly*. This is still a rather vague definition. But the notion of a topic is also vague in the human mind and usually depends on the subject. During the fourth step a similar de-duplication happens for answers to similar questions. Finally those answers are ranked according to correctness, completeness and quality.

4 Question Aggregation

The goal of the algorithm proposed in this paper is to assign similarity scores to question post pairs, such as the following about a specific problem with the Java programming language:

I can add nodes to my JTree, but after expanding a node, I can not add any child nodes to that node, or at least they dont show on the screen? *Actually I can use the .add(new DefaultMut ableTreeNode(„Label“)); to add a node to a parent node, but the new child node doesn‘t show on the display of the JTree?*	*I am having trouble displaying a JTree with newly added nodes.<br />* *...* *Using the same exact routine as above - I add additional new children to the root, the new children do not display in the Jtree.* *...*

Our semantic similarity detection approach starts with a training phase using a large corpus of questions. The initial step is to calculate a co-occurrence matrix for terms occurring together in the same question. To select relevant terms each question is preprocessed with a stemmer and a stop word filter. Multiple co-occurrences are counted multiple times. The matrix is a representation of typical word relations. We consider two factors for the similarity calculation of two questions. The next paragraphs present a formalized notation. An example will be provided afterwards for easier understanding.

The first factor is the relation of matching term co-occurrences from both questions. At first an importance measure is calculated for each co-occurrence, based on the value in the co-occurrence matrix M, according to Equation 1. The value of i for a term co-occurrence c and a question Q is the value of both co-occurring terms $c(1)$ and $c(2)$ normalized by the maximum co-occurrence value for two terms in question Q.

$$i(c,Q) = \frac{M_{c(1),c(2)}}{max(M_Q)} \quad (1)$$

The result of *i* is high for co-occurrences which are frequent in the training set and low for infrequent ones. That way all co-occurrences are ranked in order of frequency for a question, with values ranging from 0 to 1.

The relation of two co-occurrences is calculated with Equation 2. It builds the ratio of the smaller importance value *i* to the larger one. That way the domain of *rel* is (0,1]. For frequent co-occurrences according to the co-occurrence matrix *M rel* takes on higher values than for infrequent ones.

$$rel(i_1,i_2) = \left\{ \begin{array}{ll} \frac{i_1}{i_2} & \text{, if } i_1 < i_2 \\ \frac{i_2}{i_1} & \text{, otherwise} \end{array} \right\} \quad (2)$$

Equation 2 in its pure form is only applicable to two co-occurrences with the same terms. This means that both terms *c*(1) and *c*(2) must be equal for both co-occurrences with importance values i_1 and i_2 to make them comparable. However as already mentioned the relations between terms are more complex on a semantic level. That is why a second, diminishing factor is added for co-occurrences with terms having a similar meaning. This factor results from semantic similarities between the terms of two co-occurrences according to the word database WordNet [21]. There are different researchers who provided different similarity measures based on word relations from WordNet [23][13][15][10]. Since it was shown to produce high quality results, the measure provided by Lin [13] will be used. Its domain is defined as a value of 1 for equal words and a value close to 0 for very unrelated words. For two co-occurrences *lin* is calculated as the average of all four possible relations. Since *lin* never reaches 0 a threshold is defined to filter out all similarities below a certain value. All values of *lin* below this threshold are set to 0.

These two factors result in Equation 3 to calculate the similarity between two questions Q_1 and Q_2 as the sum of the two presented factors over all co-ocurrences from Q_1 and Q_2.

$$sim(Q_1, Q_2) = \frac{\sum_{c_1 \in Q_1} \sum_{c_2 \in Q_2} (rel(i(c_1, Q_1), i(c_2, Q_2)) \cdot lin(c_1, c_2))}{|Q_1| \cdot |Q_2|} \quad (3)$$

Equation 3 also contains a factor normalizing the result with the product of the lengths of the two questions.

For an example consider the following two term sets, resulting from two questions after stop word filtering and stemming:

Q_1: awt, swing, concept, awt	Q_2: concept, awt, swing, component

The co-occurrence matrix presented in Table 1 is based on those two and some additional not shown questions with additional words. The value in the first row, fourth line results from the term "swing" occurring three times together with "awt", for instance. The most important co-occurrence for the second question is "swing" and "component", since among the co-occurrences from that question it has the largest value in the co-occurrence matrix. Thus all other values are normalized with 6 and the importance of "swing" and "awt" for question two computes to 3/6. Inserting these values into Equation 3 results in Equation 4. Table 1 also shows all the importance values for all co-occurrences of Q_1 and Q_2.

$$\begin{aligned} sim(Q_1, Q_2) &= \frac{\left(\frac{2/7}{2/6} + \frac{3/7}{3/6} + \frac{2/6}{2/7}\right) \cdot 1.0 + \left(\frac{1/7}{3/6} + \frac{1/6}{4/7} + \frac{1/6}{4/7}\right) \cdot 0.9}{4 \cdot 5} \\ &= 0.183154762 \end{aligned} \quad (4)$$

Table 1: Co-occurrence matrix and importance values i for Q_1 and Q_2.

	awt	concept	component	swing	understand	comprehend
awt	9	3	2	3	1	3
Q_1			(2/6)	(3/6)		(3/6)
Q_2		(1/7)	(2/7)	(3/7)	(1/7)	
concept	3	18	3	7	2	8
Q_1						
Q_2			(2/7)	(7/7)	(2/7)	
component	2	3	13	6	4	1
Q_1				(6/6)		(1/6)
Q_2				(6/7)	(4/7)	
swing	3	7	6	19	4	1
Q_1						(1/6)
Q_2					(4/7)	
understand	1	2	4	4	11	1
comprehend	3	8	1	1	1	12

5 Evaluation

This section shows the performance of the Effingo similarity detection algorithm. It focuses on two areas. First it explores the influence of the threshold value for the semantic similarity, as proposed in the previous section and second it compares the results generated by the similarity measure to human assessments.

Dataset

The dataset consists of 315 question posts with the shared domain of being about Java programming. We started with a collection of 15 seed questions and added near-duplicates using Google and diverse forum search engines. We also added non near-duplicates from the same forums, just using the most recent threads. That way we created 15 sets of 20 posts for each seed question.

Three human annotators evaluated each of the 15 sets and created a ranking of how similar they thought each question to be to the seed question. They also should mark a question, if they think the question is the same or at least partly the same as the seed question. This ground truth was used for all further evaluations presented in this section. General agreement between the human annotators was quite good. All of them recognized nearly all of the semantically similar questions and ordered them at the top of the list. However the absolute ranking varied heavily from annotator to annotator. It was noticeable that all annotators tended to move shorter questions on higher ranks. In addition some near duplicates were not recognized correctly. Such cases occurred for very long and complicated questions, containing the actual question only at the end. The evaluation results are calculated using each annotator's assessment separately, averaging the individual results to generate the final values.

Evaluation Measures

The results are evaluated using three measures as proposed by Vaughan et al. [19]. They are called Precision, Recall and F_1 score, even though they are not exactly the same as proposed by the Message Understanding Conference (MUC) for the evaluation of classification results.

$$p = 1 - \frac{6 \cdot \sum_i [rg(x_i) - rg(y_i)]^2}{n(n^2 - 1)} \tag{5}$$

Precision is calculated using the simplified Spearman coefficient according to Equation 5. The value of *n* is the amount of posts in the list. So it is always 20 for our evaluation. The result of *rg*(*x*) is the rank of the post x either in the ground truth or as provided by the algorithm. The Spearman coefficient takes on values between -1 for no agreement between the ranking from the gold standard and the human labeler up to 1 for absolute agreement.

Recall is calculated according to [19] by counting the correct results from the top *k* results, where *k* is the amount of expected results. This means recall is 1 if the top *k* actually are the expected semantically similar questions and decreases to 0 if none of the top k results is a relevant result.

F_1 score is the harmonic mean of Precision and Recall as proposed for MUC-1 [8].

Threshold Evaluation

As a first step we tried to find the optimal threshold for semantic similarity. A low threshold would include very far semantically related terms, while a high threshold would reduce the approach to a simple syntactic co-occurrence comparison. Figure 2 shows the development of the F_1 score on our dataset, using different threshold values. The optimal value for our dataset is at approximately 0.74, shown by the vertical black line. This value is used for the following experiments.

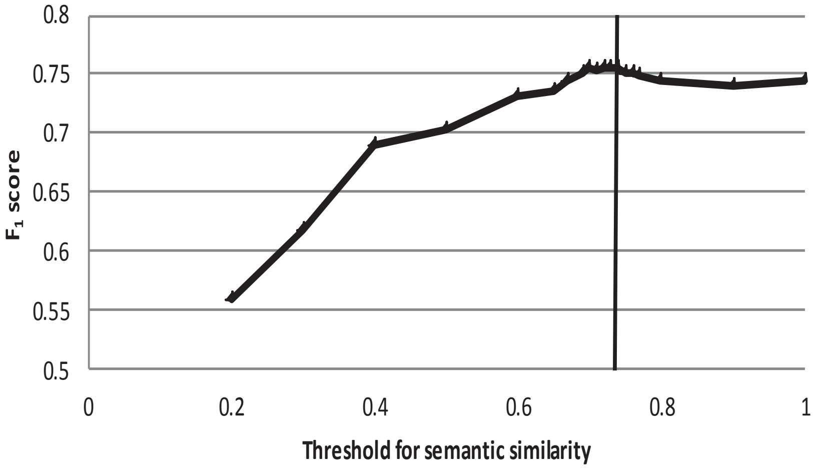

Figure 2: Development of F_1 score for different thresholds

The hump around the maximum has the following explanation. For low thresholds, words with low relation scores are considered as similar. However, words such as “help” and “use” might occur by chance, reducing result quality. At around 0.74 only direct hypernyms and synonyms are considered. At around 0.75 only synonyms are considered and F_1 score drops again.

Results

The final results of the algorithm proposed for this work were compared to three other approaches. The semantic baseline calculates semantic similarity between all words from Q_1 to Q_2 according to Lin [13] and adds them together. In addition two syntactic approaches were compared. The syntactic baseline is a simple tf-idf approach. It creates a term vector for Q_1 and Q_2 and compares them using their scalar product. We also tried a well established Information Retrieval framework; Apache Lucene [21]. In this approach the 20 questions from each group are put to a Lucene index, which is queried using the seed question. Table 2 visualizes the results of those three approaches and the Effingo approach.

Table 2: Results for applying different algorithms to the evaluation dataset.

	Spearman/Precision	Recall	F_1 score
Semantic Baseline	0.133	0.497	0.210
Syntactic Baseline	0.589	0.790	0.675
Effingo	0.670	0.864	0.755
Lucene	0.667	0.867	0.754

As expected the semantic baseline and the tf-idf approach are not even close to the Effingo approach. The semantic baseline even resembles an equal distribution of relevant and non relevant questions, which should have a Spearman coefficient of 0.135. Both approaches – the baseline and the tf-idf approach – suffer from the problem that long texts tend to have a higher chance of having relevant terms than short texts

However, the purely syntactic Lucene algorithm shows similar results than the Effingo approach. This probably results from the way the ground truth was created. Since a search engine was used to find the near-duplicates in the first place, it is not surprising that another search engine such as Lucene is capable of finding those texts again. However, the Effingo approach was capable of finding a significant portion of the relevant texts as well. It identified 128 of the 150 near-duplicates correctly. Those which were not found, were usually replaced by a false positive with a similar length than the seed question, which often also contained some relevant words and thus was capable of replacing the weakest correct near-duplicates. In addition the hump in Figure 2 shows that semantic relations do indeed have an influence on the quality of similarity detection, at least compared to a pure syntactic co-occurrence measure (high threshold). Also, the Effingo approach provides a normalized score in contrast to the Lucene approach, which is comparable across different queries. This is important for all algorithms build on top of it.

The Effingo approach suffers from the limitation, that WordNet is a generic knowledge base, which knows nothing about the target domain. This makes the approach domain independent on the one hand, losing much information provided by domain dependent terms on the other. So instead of associating Java with a programming language it might associate it with coffee or the term “island”. Additionally it still suffers from the problem that long texts tend to swallow shorter texts because of chance matches.

6 Summary and Future Work

This paper explained an approach to find semantic near-duplicate questions from social media sites. It is based on the idea that co-occurrences of words are important to identify two differently phrased questions with the same content. In addition it incorporates the knowledge base WordNet, to identify similarities between not absolutely matching terms.

The paper shows that the approach is better than simple baseline approaches and reaches the performance of current state of the art information retrieval systems.

As a next step we plan to merge the presented semantic approach with a syntactic one such as Lucene, to improve on results from both approaches. The goal is to create a system to aggregate questions from all over the Web and provide users with an aggregated view on a question and its answers. We also intend to provide a better ground truth, containing pairs that get a higher benefit from semantic relations and contain fewer syntactic matches. Finally we are going to address the problem, that long texts swallow small ones, with adaptations to our scoring equations.

Acknoledgements

The research shown in this paper was carried out as part of the diploma thesis of Christian Hensel.

References

[1] Allan J., Carbonell J., Doddington G., Jamron J. P., Yang Y., Topic detection and tracking pilot study: Final report, 1998

[2] Allan J., Lavrenko V., Swan R., Explorations within topic tracking and detection, 2002

[3] Bengel J., Gauch S., Mittur E., Vijayaraghavan R., Chattrack: Chat room topic detection using classification, 2004

[4] Chen H., Ku L., An NLP & IR approach to topic detection, 2002

[5] Cong G, Wang L., Lin C. Y., Song Y. I., Sun Y., Finding question-answer pairs from online forums, 2008

[6] Dharanipragada S., Franz M., McCarley J. S., Ward T., Zhu W.-J. Segmentation and detection at IBM: hybrid statistical models and two-tiered clustering, 2002

[7] Eichmann D., Srinivasan P., A cluster-based approach to broadcast news, 2002

[8] Grishman R., Sundheim B., Message understanding conference-6: A brief history, 1996.

[9] Hong L., Davison B., A classification-based approach to question answering in discussion boards, 2009

[10] Leacock C., Chodorow M. Miller G. A., Combining local context and WordNet sense similarity for word sense identification, 1998

[11] Leek T., Schwartz R., Sista S., Probabilistic approaches to topic detection and tracking, 2002

[12] Levow G., Oard D., Signal boosting for translingual topic tracking: Document expansion and n-best translation, 2002

[13] Lin D., An information-theoretic definition of similarity, 1998

[14] Apache Lucene, https://lucene.apache.org/, Last visited on 04/15/2013

[15] Resknik R. Using information content to evaluate semantic similarity, 1995

[16] Pretzsch S., Muthmann K., Schill A., FODEX – Towards Generic Data Extraction from Web Forums, 2012

[17] Schultz J., Liberman M., Towards a "Universal dictionary" for multi-language information retrieval applications, 2002

[18] Tang X., Approach to detection of community's consensus and interest, 2008

[19] Vaughan L. New measurements for search engine evaluation proposed and tested, 2003

[20] Wang H., Huang T., Guo J., Li S., Journal Article Topic Detection Based on Semantic Features, 2009

[21] Miller G. A., WordNet: A Lexical Databse for English, 1995

[22] Wu Z., Li C., Topic Detection in Online Discussion using Non-negative Matrix Factorization, 2007

[23] Wu Z., Palmer M., Verbs semantics and lexical selection, 1994

[24] Xu G., Ma W., Building implicit links from content for forum search, 2006

[25] Yamron J., Gillick L., van Mulbregt P., Knecht S., Statistical Models of Topical Content, 2002

[26] Yang C., Ng T., Analyzing Content Development and Visualizing Social Interactions in Web Forum, 2008

[27] Yang Y., Carbonell J., Brown R., Lafferty J., Pierce T., Ault T., Multi-strategy Learning for Topic Detection and Tracking, 2002