Boolean Queries for News Monitoring: Suggesting new query terms to expert users

Suzan Verberne
Radboud University
Nijmegen, the Netherlands
s.verberne@cs.ru.nl

Thymen Wabeke
TNO
The Hague, the Netherlands
thymen.wabeke@tno.nl

Rianne Kaptein
TNO
The Hague, the Netherlands
rianne.kaptein@tno.nl

Abstract

In this paper, we evaluate query suggestion for Boolean queries in a news monitoring system. Users of this system receive news articles that match their running query on a daily basis. Because the news for a topic continuously changes, the queries need regular updating. We first investigated the users' working process through interviews and then evaluated multiple query suggestion methods based on pseudo-relevance feedback. The best performing method generates at least one relevant term among 5 suggestions for 25% of the searches. We found that expert users of news retrieval software are critical in their selection of query terms. Nevertheless, they judged the demo application as clear and potentially useful in their work.

1 Introduction

LexisNexis Publisher[1] is an online tool for news monitoring. Hundreds of organizations in Europe and the US use the tool to collect news articles relevant to their work. An organization typically monitors multiple topics. For monitoring the news for a user-defined topic, LexisNexis Publisher takes a Boolean query as input, together with a selection of news sources and a date range. Two example queries can be found in Table 1.

Table 1: Examples of Boolean queries

Topic	Boolean query
Products in the News	"Output Campaign Manager" or "TransPromo" or "Output Wrap Envelope" or "Adsert" or "OptiMail" or "ePriority" or "output Address Direct" or "PredictionPro" or "offmydesk"
Diversity	Diversity /2 inclusion OR "equal employment" or discrimination or harassment or race or gender or religion or "national origin" or disability

Interviews with users of LexisNexis Publisher indicate that noise in the set of retrieved documents is not very problematic because the user has the option to disregard irrelevant documents in the selection, thereby controlling precision. Recall is more difficult to control because the user does not know the documents that were not found. For the user, it is important that no relevant news stories are missed. Therefore, the query needs to be extended when there are changes to the topic. This can happen when new terminology becomes relevant for the topic (e.g. 'wolf' for the topic 'biodiversity'), when there is a new stakeholder (e.g. the name of the new minister of economic affairs for the topic 'industry and ICT') or when new geographical names are relevant to the topic (e.g. 'Lesbos' for the topic 'refugees'). The goal of the current work is to support users of news monitoring appli-

In: M. Martinez, U. Kruschwitz, G. Kazai, D. Corney, F. Hopfgartner, R. Campos and D. Albakour (eds.): Proceedings of the NewsIR'16 Workshop at ECIR, Padua, Italy, 20-March-2016, published at http://ceur-ws.org

[1] http://www.lexisnexis.com/bis-user-information/publisher/

cations by providing them with suggestions for new query terms in order to retrieve more relevant news articles.

Our intuition is that documents that are relevant but *not* retrieved for the current query have similarities with the documents that *are* retrieved for the current query. Therefore, our approach to query suggestion is to generate candidate query terms from the set of retrieved documents.

In this paper, we present the results of a user study in which we evaluate our methodology for query term suggestion with 9 expert users of LexisNexis Publisher. We first conducted interviews with the users to collect their wishes and needs. Then we developed a demo application for news retrieval with query term suggestion functionality. We used this application to evaluate our approach and compare 12 different methods for query term suggestion.

2 Related work

The task of spotting novel terms in a news stream is related to research on topic detection and tracking (TDT) which has its roots in the 1990s [2, 1]. TDT aims to automatically detect new topics or events in temporally-ordered news streams, and to find new stories on already known topics. The functionality of LexisNexis Publisher is related to news tracking in TDT: the topic is given (in the form of a query) and the tool is expected to find relevant new stories in the news stream [14]. More recent work on TDT is directed at topic tracking in microblog data (Twitter) [10, 5]. Microblog data, like news data, is temporally ordered data that continuously changes.

Our approach to query suggestion – generating candidate query terms from the set of retrieved documents – is related to pseudo-relevance feedback [3], a method for query expansion that assumes that the top-k retrieved documents are relevant, extracting terms from those documents and adding them to the query. Pseudo-relevance feedback has been applied to microblog retrieval, expanding the user query with related terms from retrieved posts to improve recall [6, 8]. It is important to take into account that the language use around a topic continuously evolves when selecting terms from Twitter and news data. One option is to give a higher score to terms that are temporally closer to query time [6]. Our approach to query term suggestion is related to this idea: we aim to find the terms that are prominent in the most recent news articles on a topic.

There are two key differences between pseudo-relevance feedback and our approach: First, instead of adding terms blindly, we provide the user with suggestions for query adaptation. Second, we deal with Boolean queries, which implies that we do not have a relevance ranking of documents to extract terms from. This means that the premise of 'pseudo-relevance' may be weak for the set of retrieved documents.

3 Interviews with expert users

We conducted interviews with three experienced users of LexisNexis Publisher to get to know their way of working, their priorities and their wishes for query assistance. The following paragraphs summarize the insights obtained during these interviews.

Way of working. Queries are not changed frequently; most attention is paid to the initial query. Formulating this query takes several hours up to a whole day. Query constructions with Boolean operators are often re-used, for example to exclude specific sources or newspaper sections. If a query gives too much noise, exclusions are added (using the 'NOT' operator). If a query gives too few results, new terms are added (with the 'OR' operator). Changes that are made a later stage are often changes in person and place names. Some customers have difficulties formulating good Boolean queries. These customers make use of information specialist at LexisNexis to formulate their queries.

Priorities. The experts we interviewed use LexisNexis Publisher to create newsletters for their organization. Typically, they review all the retrieved articles before deciding which are included in the newsletter. This selection is based on redundancy and relevance; in case of overlapping news articles, the longest story from the most reliable source is selected. This is done manually, as it allows users to control the precision of the news articles included in the newsletter. The users indicate that for this reason, it is especially important that no relevant documents are missed by the search. Noise in the result set is not so much an issue; if half of the retrieved articles is relevant, the users are satisfied.

Wishes for query assistance. Users indicate that assistance in query formulation could be helpful, not only when adapting existing queries, but especially when formulating new queries. The users mention assistance in the form of: (a) suggestions of new query terms; (b) suggestions for deleting query terms that give too much noise; (c) suggestions for deleting query terms that give very few results. Of these three tasks, we concentrated on the first: suggesting potential new query terms. One requirement posed by the users is that the user still has full control over the query. Terms should not be added blindly, but be presented as suggestions.

4 Methodology

Our approach to query suggestion is to generate candidate query terms from the set of retrieved documents.[2] The central methodology needed for generating terms from a document collection is term scoring; each candidate term from the document collection is assigned a score that allows for selecting the best – most descriptive – terms. The term scoring methods that we use are defined below.

Problem definition. We have a text collection D (the 'foreground collection') consisting of one or more documents. Our goal is to generate a list of terms T with for each $t \in T$ a score that indicates how *descriptive* t is for D. Each t is a sequence of n non-stopwords; we use $n = \{1, 2, 3\}$ in our experiments.

In most term scoring methods, descriptiveness is determined by comparing the relative frequency of t in the foreground collection D to the relative frequency of t in a background collection. For a given Boolean query, we retrieve the result set R_{recent}, which is the set of articles published in the last 30 days, and the result set R_{older}, which is the set of articles published 60 to 30 days ago.

Methods for generating descriptive terms. We compare three methods for generating the most relevant query terms (see Figure 1 for a schematic overview):

A. Return the top-k terms from T_1, generated using R_{recent} as the foreground collection and a generic news corpus as background collection;[3]

B. Return the top-k terms from T_2, generated using R_{recent} as foreground collection and R_{older} as background collection;

C. First generate T_3, using R_{older} as foreground collection and the generic news corpus as background collection. Then return the top-k terms from the set $\{t : t \in T_1 \wedge t \notin T_3\}$ (all terms from T_1 that are not in T_3).

Term scoring algorithms. We implemented four different term scoring algorithms from the literature that we compare for the task of generating potential query terms from the set of retrieved documents:

- Parsimonious Language Models (PLM) [4], designed for creating document models in Information Retrieval. In PLM, the term frequency for each t in D is weighted with the frequency of t in the background collection using an expectation-maximization algorithm;
- Kullback-Leibler divergence for informativeness and phraseness (KLIP) [12]. Informativeness is determined by comparing the relative frequency of t in D to the relative frequency of t in the background collection. Phraseness is determined by comparing the frequency of t as a whole to the frequencies of the unigram that the n-gram t is composed of; Informativeness of t and Phraseness of t are summed to obtain a relevance score for t.
- Frequency profiling (FP) [11], designed for contrasting two separate corpora. This method uses a log-likelihood function based on expected and observed frequencies of a term in both corpora (the foreground and background collections);
- Co-occurrence Based χ^2 (CB) [7], which determines the relevance of t in the foreground collection by the distribution of co-occurences of t with frequent terms in the collection itself. The rationale of this method is that no background corpus is needed because the set of most frequent terms from the foreground collection serves as background corpus.

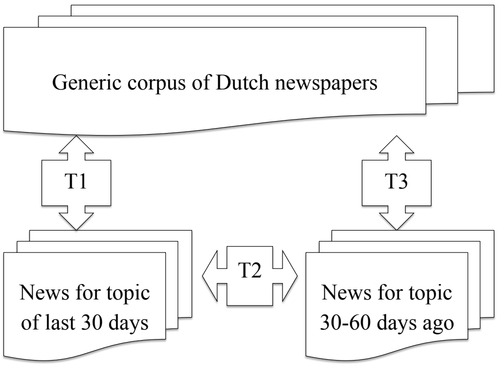

Generic corpus of Dutch newspapers
T1
T3
T2
News for topic of last 30 days
News for topic 30-60 days ago

Figure 1: Schematic view of how the term lists are generated. The query suggester returns one of three term lists to the user: $A = T_1$; $B = T_2$ and $C = \{t : t \in T_1 \wedge t \notin T_3\}$.

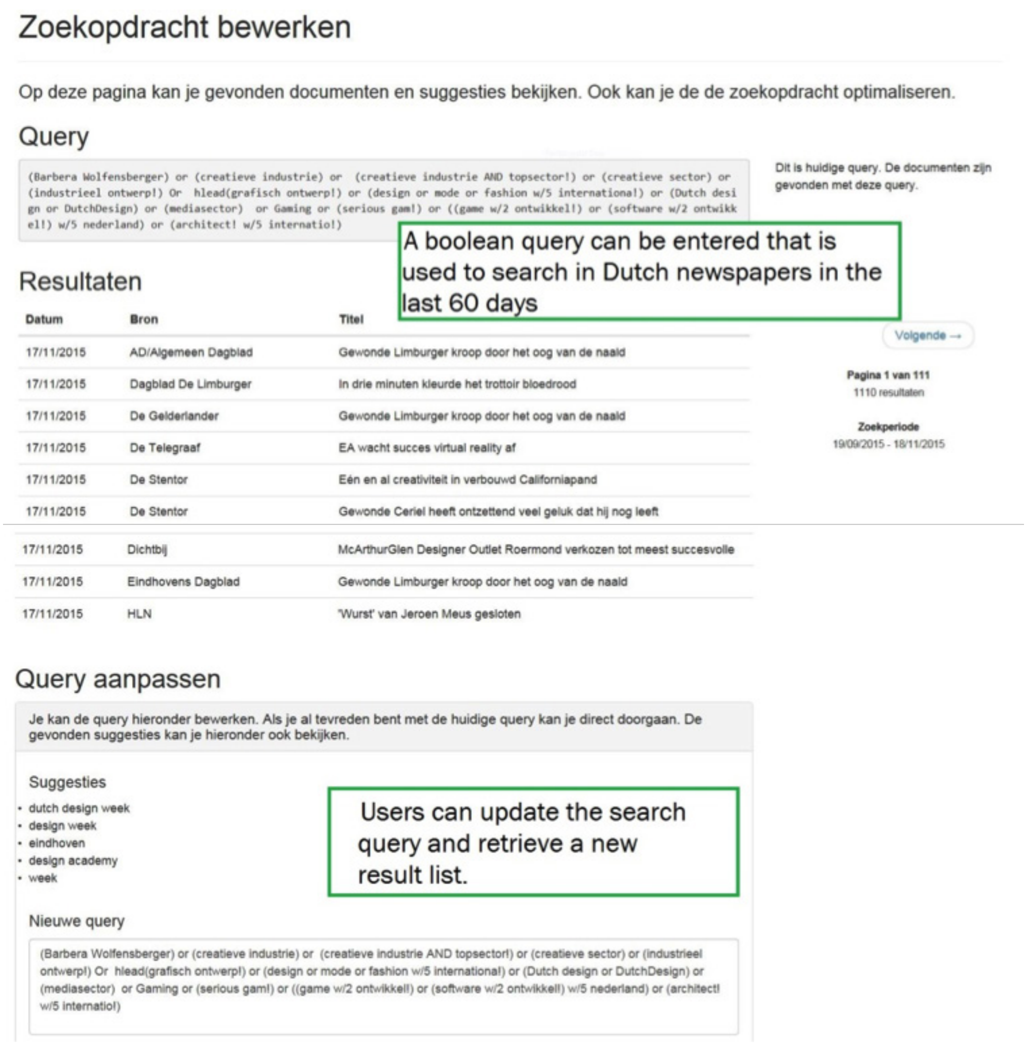
For one query and the corresponding retrieved documents, we generate twelve lists of potential query terms: three different approaches (A–C) with four term scoring algorithms.

5 Experiment and results

We collected feedback from expert users of LexisNexis Publisher to determine the best method for generating term suggestions. For this purpose, we developed an external demo application for news retrieval from the LexisNexis collection that includes query term suggestion functionality. Note that the query term suggestion functionality was not integrated in the existing LexisNexis search interface, but implemented as a standalone web application. Figure 5 shows a screen-

[2] A query term may consist of multiple words.

[3] We used the newspaper section from the Dutch SoNaR-corpus [9], 50 Million words in total. Available at http://tst-centrale.org/producten/corpora/sonar-corpus/6-85



Zoekopdracht bewerken

Op deze pagina kan je gevonden documenten en suggesties bekijken. Ook kan je de de zoekopdracht optimaliseren.

Query

```
(Barbera Wolfensberger) or (creatieve industrie) or  (creatieve industrie AND topsector!) or (creatieve sector) or
(industrieel ontwerp!) Or  hlead(grafisch ontwerp!) or (design or mode or fashion w/5 international!) or (Dutch desi
gn or DutchDesign) or (mediasector)  or Gaming or (serious gam!) or ((game w/2 ontwikkel!) or (software w/2 ontwikk
el!) w/5 nederland) or (architect! w/5 internatio!)
```

Dit is huidige query. De documenten zijn gevonden met deze query.

A boolean query can be entered that is used to search in Dutch newspapers in the last 60 days

Resultaten

Datum	Bron	Titel
17/11/2015	AD/Algemeen Dagblad	Gewonde Limburger kroop door het oog van de naald
17/11/2015	Dagblad De Limburger	In drie minuten kleurde het trottoir bloedrood
17/11/2015	De Gelderlander	Gewonde Limburger kroop door het oog van de naald
17/11/2015	De Telegraaf	EA wacht succes virtual reality af
17/11/2015	De Stentor	Eén en al creativiteit in verbouwd Californiapand
17/11/2015	De Stentor	Gewonde Ceriel heeft ontzettend veel geluk dat hij nog leeft
17/11/2015	Dichtbij	McArthurGlen Designer Outlet Roermond verkozen tot meest succesvolle
17/11/2015	Eindhovens Dagblad	Gewonde Limburger kroop door het oog van de naald
17/11/2015	HLN	'Wurst' van Jeroen Meus gesloten

Pagina 1 van 111
1110 resultaten

Zoekperiode
19/09/2015 - 18/11/2015

Query aanpassen

Je kan de query hieronder bewerken. Als je al tevreden bent met de huidige query kan je direct doorgaan. De gevonden suggesties kan je hieronder ook bekijken.

Suggesties

- dutch design week
- design week
- eindhoven
- design academy
- week

Users can update the search query and retrieve a new result list.

Nieuwe query

```
(Barbera Wolfensberger) or (creatieve industrie) or  (creatieve industrie AND topsector!) or (creatieve sector) or (industrieel
ontwerp!) Or  hlead(grafisch ontwerp!) or (design or mode or fashion w/5 international!) or (Dutch design or DutchDesign) or
(mediasector)  or Gaming or (serious gam!) or ((game w/2 ontwikkel!) or (software w/2 ontwikkel!) w/5 nederland) or (architect!
w/5 internatio!)
```

Figure 2: A screen shot illustrating the functionality of the demo application for query term suggestion.

shot of the demo application.[4] The user interface is in Dutch. In the top part of the screen ('Zoekopdracht bewerken' – 'Edit search'), the user sees the current query and the results ('Resultaten') retrieved for that query. In total, 1110 results were retrieved for this query. In the bottom part of the screen ('Query aanpassen' – 'Adapt query'), the user sees a list of term suggestions. This example illustrates the final functionality, in which only the 5 suggestions by the best performing method are shown. In the experimental setting, the user saw a pool of 10–25 terms from different methods.

5.1 Evaluation design

The query suggestion software was evaluated by 9 individual users of LexisNexis Publisher. A 2-hour evaluation session was organized for each participant. The interviews described in Section 3 revealed that queries change more frequently when they are novel. Therefore, each participant was asked to perform two different tasks with the assistance of our demo application during the evaluation session. In the first task, the participant is asked to update a query that is already being used by his company. In the second task, the participant designs a new query for a topic of which they received a short topic description.

The initial (existing or new) Boolean query is issued in LexisNexis Publisher through its API, searching in Dutch newspapers of the last 60 days (the maximum posed by the API). The titles and abstracts of the matching news articles are shown in a result list (in chronological order) and a list of query term suggestions is presented. The participant reviews the set of retrieved documents and improves the query by adding and/or removing terms, optionally using a term from

[4] A video demonstrating the demo application can be viewed here: https://youtu.be/4yIYpvHVugQ

the suggestions. Subsequently, the updated query is issued and the query can be improved again. In both tasks, the participant was asked to review and update the query up to a maximum of five iterations. After the complete evaluation session, the participants filled in a post-experiment questionnaire, in which they could provide additional comments.

5.2 Data

The participants issued 83 searches in total. The Boolean queries are long: 45 terms on average. Terms can be single words or phrases (multi-word terms), and they are combined with Boolean operators. We used the LexisNexis Publisher API to retrieve documents (news articles) published in the last 60 days. On average, $1,031$ documents were retrieved per query (ranked by date), with an average length of 63 words. The short document length is caused by the API allowing us to extract only the summary of the news article, not the full text. This means that the size of the subcollection from which potential new query terms are extracted for a query is on average $1,031*63 = 64,953$ words.

We created a pool of terms from the 12 (3 approaches * 4 term scoring algorithms) term lists per topic. We assume that in a real application, the query suggestion software would show five candidate terms to the user, and we want to be able to evaluate these 5 suggestions for each method. Therefore, the top 5 terms from each term list were added to the pool. The maximum number of terms in a pool is 60 (12*5) but in reality there is quite some overlap: the number of terms per pool is between 10 and 25. For each query, the participants were presented with this pool of 10–25 terms. The terms were ranked by the number of top-5 lists they appear in: the terms that were extracted by most methods were ranked on top of the pool.

5.3 Experimental Results

The selection of query terms and the relevance judgments for the suggested terms in the pool allow us to evaluate and compare the methods. For each method, we have judgments for the 5 highest scoring terms. We count how often one of these terms was selected by a participant, and how often at least one of these terms received a relevance rating of at least 4. The results are in Table 2 and Table 3. The results for the best performing methods (method A with either FP or KLIP as term scoring algorithm, or method C with KLIP) are marked with boldface in the tables. With these methods, participants selected a term from the top-5 suggestions for 13% of the searches, and judged at least one term from the top-5 suggestions as relevant (relevance score >= 4) for 25% of the searches.

Table 2: Results per method in terms of 'selected-success-rate': the percentage of searches for which participants added a term from the top-5 to the query.

	CB	FP	KLIP	PLM
$A = T_1$	10%	**13%**	11%	11%
$B = T_2$	10%	7%	6%	6%
$C = \{t : t \in T_1 \wedge t \notin T_3\}$	10%	0%	11%	11%

Table 3: Results per method in terms of 'relevant-success-rate': the percentage of searches for which participants judged at least a term from the top-5 as relevant (relevance score >= 4).

	CB	FP	KLIP	PLM
$A = T_1$	14%	**24%**	**25%**	20%
$B = T_2$	14%	11%	13%	5%
$C = \{t : t \in T_1 \wedge t \notin T_3\}$	14%	11%	**25%**	20%

The average rating given to the terms in the pool was low: 1.36 on a 5-point scale.

Further analysis of the results showed that the term suggestions were noisy because the sets of retrieved documents are noisy. The Boolean queries return a large set of documents (more than a thousand on average for the last 60 days), without any relevance ranking. The interviews with the users indicated that this is not a problem for the users (because they filter the news items for the newsletter), but it turns out to be a problem for the extraction of relevant terms. In other words, the premise of 'pseudo-relevance' does not hold for Boolean retrieval, and this hurts the quality of query term suggestion based on retrieved documents.

5.4 Qualitative feedback

In the post-experiment questionnaire, participants indicated that the demo application was clear and intuitive (median score of 4 on a 5-point scale for the statement 'the web application is clear'). Half of the participants would be interested in using the tool. However, they felt that the quality of the terms should be improved for the application to be really useful. Suggestions that were provided by the users included:

- Do not to suggest terms that are already covered by wildcards in the query. We improved this in the final version of the demo application.
- Terms that occur in important parts of the text should be more relevant. In fact, this was already taken into account because the API only allowed us to access the abstracts of the documents.
- Multi-word terms should not be suggested. This comment appeared to be in contrast with the users' term selections: of the selected terms by the users (15), the majority (12) are multi-words.

- Add suggestions for the use of Boolean operators. This was beyond the scope of the current project, which focused on term suggestion.

6 Conclusions

The results of our user experiment show that with the best performing method, participants selected a term from the top-5 suggestion list for 13% of the topics, and judged at least one term as relevant for 25% of the topics. Inspection of the results and the post-task questionnaire revealed that the term suggestions are noisy, mainly because the set of retrieved documents for the Boolean query is noisy. We expect that the use of relevance ranking instead of Boolean retrieval, and a post-filtering for noisy terms, will give better user satisfaction.

The relevance judgments for the suggested terms are low compared to another application area for term extraction that we addressed in previous work with the same methodology, namely author profiling [13]. This can partly be explained by the noise in the set of retrieved documents (irrelevant documents lead to irrelevant terms), but may also be caused by expert users of news retrieval software being critical in their selection of query terms. This shows that it is valuable to evaluate query suggestion technology with real users.

References

[1] Allan, J.: Topic detection and tracking: event-based information organization. Volume 12. Springer Science & Business Media (2002)

[2] Allan, J., Papka, R., Lavrenko, V.: On-line new event detection and tracking. In: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, ACM (1998) 37–45

[3] Cao, G., Nie, J.Y., Gao, J., Robertson, S.: Selecting good expansion terms for pseudo-relevance feedback. In: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, ACM (2008) 243–250

[4] Hiemstra, D., Robertson, S., Zaragoza, H.: Parsimonious language models for information retrieval. In: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, ACM (2004) 178–185

[5] Lin, J., Snow, R., Morgan, W.: Smoothing techniques for adaptive online language models: topic tracking in tweet streams. In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM (2011) 422–429

[6] Massoudi, K., Tsagkias, M., de Rijke, M., Weerkamp, W.: Incorporating query expansion and quality indicators in searching microblog posts. In: Advances in Information Retrieval. Springer (2011) 362–367

[7] Matsuo, Y., Ishizuka, M.: Keyword extraction from a single document using word co-occurrence statistical information. International Journal on Artificial Intelligence Tools **13**(01) (2004) 157–169

[8] Metzler, D., Cai, C., Hovy, E.: Structured event retrieval over microblog archives. In: Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics (2012) 646–655

[9] Oostdijk, N., Reynaert, M., Monachesi, P., Van Noord, G., Ordelman, R., Schuurman, I., Vandeghinste, V.: From d-coi to sonar: a reference corpus for dutch. In: LREC. (2008)

[10] Phuvipadawat, S., Murata, T.: Breaking news detection and tracking in twitter. In: Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on. Volume 3., IEEE (2010) 120–123

[11] Rayson, P., Garside, R.: Comparing corpora using frequency profiling. In: Proceedings of the workshop on Comparing Corpora, Association for Computational Linguistics (2000) 1–6

[12] Tomokiyo, T., Hurst, M.: A language model approach to keyphrase extraction. In: Proceedings of the ACL 2003 workshop on Multiword expressions: analysis, acquisition and treatment-Volume 18, Association for Computational Linguistics (2003) 33–40

[13] Verberne, S., Sappelli, M., Kraaij, W.: Term extraction for user profiling: Evaluation by the user. In: UMAP Workshops. (2013)

[14] Yamron, J., Carp, I., Gillick, L., Lowe, S., Van Mulbregt, P.: Topic tracking in a news stream. In: Proceedings of DARPA Broadcast News Workshop. (1999) 133–136