Lexical Graphs for Improved Contextual Ad Recommendation

Conference Paper · April 2009

DOI: 10.1007/978-3-642-00958-7_21 · Source: DBLP

CITATIONS
12

READS
219

4 authors:

Symeon Papadopoulos
The Centre for Research and Technology, Hellas
260 PUBLICATIONS **4,912** CITATIONS
SEE PROFILE

Fotis Menemenis
Information Technologies Institute (ITI)
6 PUBLICATIONS **36** CITATIONS
SEE PROFILE

Ioannis (Yiannis) Kompatsiaris
The Centre for Research and Technology, Hellas
1,038 PUBLICATIONS **14,475** CITATIONS
SEE PROFILE

Ben Bratu
DataDirect Networks
21 PUBLICATIONS **142** CITATIONS
SEE PROFILE

Lexical Graphs for Improved Contextual Ad Recommendation

Symeon Papadopoulos[1], Fotis Menemenis[1],
Yiannis Kompatsiaris[1], and Ben Bratu[2]

[1] Multimedia Knowledge Laboratory, Informatics and Telematics Institute
6th km Charilaou-Thermi Road, 57001, Thermi, Thessaloniki, Greece
{papadop,fotis,ikom}@iti.gr

[2] Centre for Applications Research, Motorola Labs
St. Aubin, 91193 Gif sur Yvette, France
ben.bratu@motorola.com

Abstract. Contextual advertising is a form of online advertising presenting consistent revenue growth since its inception. In this work, we study the problem of recommending a small set of ads to a user based solely on the currently viewed web page, often referred to as content-targeted advertising. Matching ads with web pages is a challenging task for traditional information retrieval systems due to the brevity and sparsity of advertising text, which leads to the widely recognized *vocabulary impedance* problem. To this end, we propose the use of lexical graphs created from web corpora as a means of computing improved content similarity metrics between ads and web pages. The results of our experimental study provide evidence of significant improvement in the perceived relevance of the recommended ads.

Key words: Online advertising, content-targeted advertising, vocabulary impedance, lexical graphs.

1 Introduction

The tremendous growth and availability of web content and services during the past years has been largely supported by revenues driven from online advertising. Contextual advertising in the form of content-targeted advertising, i.e. displaying a small set of ads together with the piece of content a user is viewing, holds a prominent place among online advertising methods due to its effectiveness in creating revenues for online content publishers. Today, the typical context-based advertising ecosystem involves the advertisers, the advertising aggregator, a network of content publishers and the web surfers. The basic principle of serving ads in such a setting is for the advertising aggregator to select the best possible ad items from its pool of ads (provided by the advertisers) given some input web page from the network of content publishers (participating in the particular ad serving ecosystem). In the case of the widely applied Pay-Per-Click (PPC) model, the advertiser needs to pay some amount each time a web

surfer clicks on one of their ads served by the ad aggregator. Subsequently, this profit is shared between the ad aggregator and the content provider.

Arguably, the success of the above scheme, which is usually quantified by means of the Click-Through-Rate (CTR), i.e. the number of user clicks on ads over the total number of displayed ads, depends on the relevance of the recommended ads to the particular piece of content under view. Since no prior information is assumed about the web surfer, the advertising aggregator has to rely solely on the content semantics of the context web page and of the ad items available in its repository in order to come up with a recommendation. In real settings, the advertising aggregator should also take into account the amount that an advertiser would be willing to pay for each user click before deciding which ads to recommend [1]. However, this paper is restricted to discussing the semantic aspect of the ad-content matching problem, i.e. how to maximize the relevance of ad recommendation given the context of a user (e.g. web page).

The main challenge faced by advertising aggregators when matching web pages with ads is the brevity and the idiosyncratic nature of the language used in ad items. This problem was first recognized by Ribeiro-Neto et al. in [2] and became known as the *vocabulary impedance* problem, pointing to the mismatch between the vocabulary of a web page and the vocabulary of an advertisement. To alleviate this problem, we present a novel contextual ad recommendation framework based on the use of lexical graphs created from web corpora. We employ graphs that carry semantic information, frequency and cooccurrence information about the terms of a web corpus from which they are created. Then, our recommendation techniques use this information as a means to reduce the vocabulary impedance between a web page and the available set of ad items.

The paper is structured as follows. The next section reviews existing work in the area of contextual advertising. Section 3 introduces the proposed ad recommendation framework and provides details on its individual components. The evaluation study comparing the proposed methods to the baseline is discussed in Section 4. Finally, the paper concludes in Section 5 with a reference to the main contributions and an outlook on the future work.

2 Background

2.1 Related Work

A number of techniques have been proposed for matching content items (web pages with textual content) with a large pool of textual advertisements. The first work addressing the problem of vocabulary impedance between advertisement and content items was presented by Ribeiro-Neto et al. [2]. The authors describe a set of techniques meant to couple the vocabulary impedance between content and ad items, based on a vocabulary expansion methodology. This involves the exploitation of the vocabulary extracted from the landing page pointed to by the advertisement and the use of a large Bayesian network trained with a significant amount of text corpora as another source of vocabulary enrichment. Further,

Lacerda et al. [3] employ a genetic programming (GP) method to associate web pages with advertisements. The proposed method aims at optimizing a fitness function by means of GP so that the most relevant advertisements end up in the top positions of the consumed web pages. Finally, Murdock et al. in [4] tackle the vocabulary impedance problem by means of machine learning techniques originating from the field of machine translation.

Another interesting group of contextual advertising systems attempt to tackle the problem of content-ad matching by refined text analysis techniques. In [5], Yih et al. propose a system that applies several feature selection techniques to extract keywords from web pages for advertising purposes. A method that uses mixtures of statistical language models to select content-relevant advertisements for personal blog pages is presented by Mishne and de Rijke [6]. Broder et al. [7] combine both the semantics (by means of a large taxonomy) and the syntax (bag of words) of advertisements and web pages to define an optimal advertisement-content matching strategy. Finally, in the work of Richardson et al. [1], the CTR of advertisement items on web pages is predicted by means of a logistic regression function that is trained using a large set of click-stream data coming from the Microsoft search engine.

To our knowledge, this is the first work where lexical graphs similar to the ones presented in [8] are used to improve the effectiveness of the contextual advertising method.

2.2 Formulation of the Ad Recommendation Problem

In the following, we formulate the problem of contextual ad recommendation. Given a set of ads $A = \{\alpha_1, \alpha_2 ... \alpha_N\}$ and a web surfer requesting a web page p, the ad recommendation task is defined as selecting a small subset of ad items $A_{rec} \subseteq A$ such that its elements are ranked first when ordered according to a given relevance measure r, which is a function of only the web page and each ad item, i.e. $r = r(p, \alpha_i)$. In order to better define the scope of the proposed system and specify evaluation measures, we are going to consider only ad items of the type *Sponsored links*. The main characteristic of this kind of ads is that they consist of purely textual elements, namely a title, a short description and a set of keywords attached by the advertiser for classification and targeting purposes.

A widely used relevance measure originating from the Information Retrieval (IR) literature is cosine similarity (COS), defined as:

$$r_{COS} = \cos(p, \alpha_i) = \frac{p \cdot \alpha_i}{|p| \times |\alpha_i|} = \frac{\sum_{k=1}^{n} w_{pk} \cdot w_{\alpha k}}{\sqrt{\sum_{k=1}^{n} w_{pk}^2} \cdot \sqrt{\sum_{k=1}^{n} w_{\alpha k}^2}} \tag{1}$$

where both the web page p viewed by the user and the arbitrary ad item a_i are represented by $p = (w_{p_1}, w_{p_2}, ..., w_{p_n})$ and $a_i = (w_{\alpha_1}, w_{\alpha_2}, ..., w_{\alpha_n})$ respectively, i.e. by the vector-space model (VSM) [9] widely used among IR practitioners. The terms of the ad item are extracted from its textual elements, i.e. its title, description and keywords.

Another popular relevance measure is the Ads-And-Keywords (AAK) measure introduced in [2]. According to this, the relevance of a_i to p is defined as:

$$r_{AAK} = AAK(p, \alpha_i) = \begin{cases} cos(p, \alpha_i) & \text{if } k_\alpha \subseteq p \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where k_α is the set of keywords which the advertiser has attached to the ad item for classification and targeting purposes.

3 System Overview

The proposed contextual advertising framework is based on the notion of lexical graphs. The graphs used within our system have the form of a network of connected words (terms), similar to the ones presented in [8]. The graphs are progressively built up through processing textual content found on the web. The co-occurrences of terms within these web corpora are exploited to build such networks of terms. The basic elements of this model, denoted by G, are the set of graph nodes (or vertices) V and the set of graph edges E, connecting pairs of nodes; in short, $G \equiv \{V, E\}$.

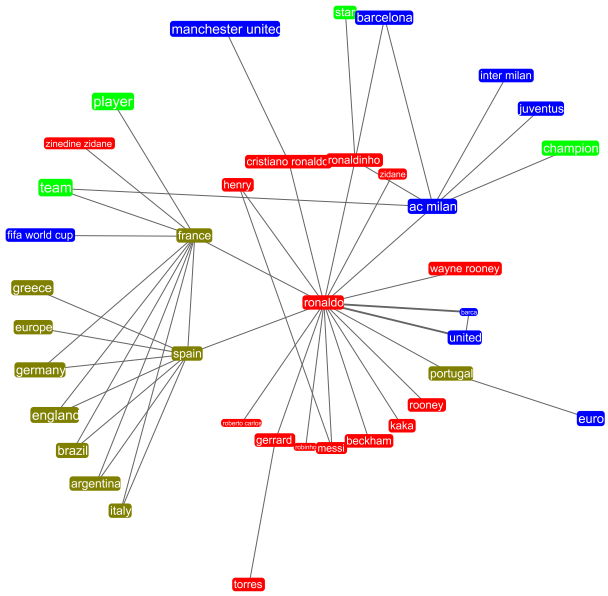

Fig. 1. A small graph excerpt created from a corpus of football related documents. Color code: Green - NN, Red - P, Blue - O, Brown - L.

The graph nodes constitute a representation scheme for the corpus terms and the graph edges express the relations (in the sense of co-occurrence) among

them. The attributes stored for each node of the graph are: the lemma of the term (string), the term frequency in the web corpus, the document frequency of the term, and the term type (e.g. whether the term is a noun (NN), a Named Entity (NE), i.e. person (P), location (L), organization (O), etc. or an adjective (ADJ)). Thus, for each node of the graph $v_i \in V$, we define four functions, namely $lemma_V(v_i) : V \rightarrow S$, $tf_V(v_i) : V \rightarrow N$, $df_V(v_i) : V \rightarrow N$, and $type_V(v_i) : V \rightarrow T \equiv \{NN, P, L, O, ADJ\}$. Each edge $e_i \in E$ of the graph connecting two terms contains the number of co-occurrences of these two terms as an attribute, i.e. the function $cooc_E(e_i) : E \rightarrow N$ is defined. In the above, S denotes the set of all strings, N is the set of the natural numbers and T is the set of the considered term types. Figure 1 presents a snapshot of a small graph excerpt created from a web corpus about football.

In the proposed system, a set of topics is defined (by the system administrator) and subsequently one graph per topic is created and maintained. There are two significant reasons for this choice: (a) terms will present different attributes and relations to other terms depending on the topic and (b) the creation and maintenance of the graphs is decomposed to independent pieces that can be handled in parallel. The system consists of two main components, (a) the lexical graph creation and maintenance component and (b) the ad recommendation component. Figure 2 presents an overview of the system.

3.1 Lexical Graph Creation and Maintenance

The lifecycle of a graph within the advertising system is handled by the graph creation and maintenance component as an offline process. The individual processing steps involved in it are described below.

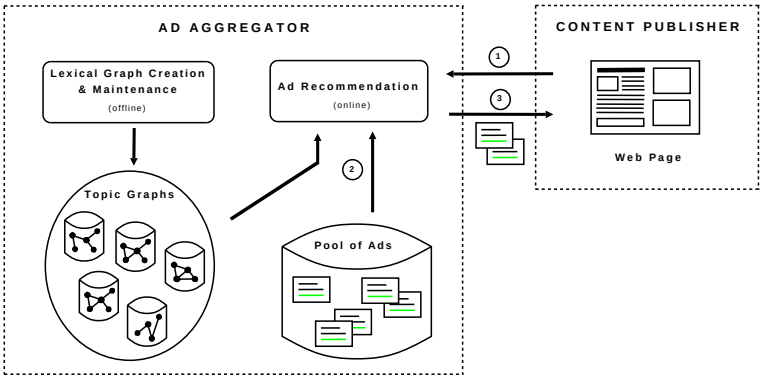

Fig. 2. Overview of the graph-based ad recommendation framework.

Web document collection. The first step in the graph creation lifecycle is the collection of web documents starting from a set of topic-related keywords. These

are submitted to a search engine and the top results are assumed to be topic-relevant and are, therefore, used to form the input web corpus. This process is repeated occasionally in order to enrich the corpus with up-to-date content.

Extraction of terms and co-occurrences from the collected documents. The HTML code of the collected web documents is parsed and the extracted full sentences are analyzed by a Natural Language Processing (NLP) module in order to derive Part-of-Speech (PoS) tags and NE types for the contained terms. Co-occurrences are calculated per sentence, i.e. two terms are considered to co-occur only when they appear in the same sentence.

Graph creation and update. After processing the collected web corpus, a set of sentences is produced, together with the terms extracted from them. Assuming that the set V_{S_I} of terms is extracted from sentence S_I, the set of all possible co-occurrences E_{S_I} between these terms is formed, resulting in the graph $G_{S_I} \equiv \{V_{S_I}, E_{S_I}\}$. For $I = 1$, the first graph $G = G_1 = G_{S_1}$ is created from S_1. For $I > 1$, an existing graph $G_{I-1} \equiv \{V_{I-1}, E_{I-1}\}$ is already available, and a graph merging operation is necessary between G_{I-1} and G_{S_I}. The graph resulting from this merging is defined as $G_I \equiv \{V_I, E_I\}$, with $V_I \equiv V_{I-1} \cup V_{S_I}$ and $E_I \equiv E_{I-1} \cup E_{S_I}$. Naturally, for a term $v_I \in V_{I-1} \cap V_{S_I}$ its term and document frequencies in the merged graph will be $tf_{V_I} = tf_{V_{I-1}} + tf_{V_{S_I}}$ and $df_{V_I} = df_{V_{I-1}} + df_{V_{S_I}}$ respectively. Similarly, for an edge $e_I \in E_{I-1} \cap E_{S_I}$ its cooccurrence frequency in the merged graph will be $cooc_{E_I} = cooc_{E_{I-1}} + cooc_{E_{S_I}}$.

Collection of graph statistics. Graph statistics are invaluable for gaining insights about the importance of terms and relations within a given graph. Minimum and maximum values as well as medians and standard deviations are calculated for the following variables: term frequency (tf), document frequency (df), co-occurrence ($cooc$) as well as node degree (deg), i.e. number of neighbors. These statistics are subsequently exploited in the graph pruning step.

Graph pruning. Pruning the graph, i.e. removing nodes and edges that are considered of minor importance for the recommendation process, is necessary for two reasons: (a) reducing the graph size, (b) eliminating "noisy" terms and relations, i.e. graph elements of circumstantial nature (e.g. they happened to appear in just one particular piece of text). Two types of pruning are applied.

Single-graph filtering. Vertex filtering is based on vertex attributes of the particular graph, e.g. tf is compared against a minimum threshold, lemma is checked against a list of common nouns, etc. Filtering of edges is based on their co-occurrence attribute.

Multi-graph filtering. The appearance of a term in more than one topic graphs indicates that the term is topic-independentSuch terms are considered irrelevant for the ad recommendation process and are therefore removed from the corresponding graphs.

3.2 Graph-based Ad Recommendation

Once a set of graphs have been created (one for each of the predefined topics), it is possible to proceed with the actual ad recommendation phase. This takes place online, in two steps: (a) analysis of input content, and (b) ad-content matching.

Analysis of input content item. As soon as the input web page is available, it is processed by the HTML and NLP analysis modules of the system and the extracted terms are provided as input to a topic classification module, which decides about the graph that should be used in the ad-content matching phase. The topic classification is performed by selecting the topic graph G_X that presents the highest similarity score with the vector of terms p extracted from the input document. More specifically, given N terms extracted from the input content, with C out of which belonging to G_X, the similarity score is defined as:

$$sim_{G_X}(p) = \frac{C}{N} \cdot \frac{\sum_{i=1}^{C} tf_{V_X}(p_i) \cdot deg_{G_X}(p_i)}{max_{V_X}(tf) \cdot max_{G_X}(deg)} \tag{3}$$

where $max_{V_X}(tf)$ is the maximum term frequency and $max_{G_X}(deg)$ is the maximum degree occurring in G_X respectively.

Ad-content matching. Once the input content has been processed, a set of terms extracted from its textual part as well as its topic are available to the ad-content matching module. Additionally, the terms of the stored ads are available to the module by tokenizing the titles and descriptions of the ad items and looking up the tokens in the respective topic graph (due to the brevity and non-conformance to syntactic and grammatical rules of the ad text, standard NLP tools fail to recognize the terms). We assume that the topic of the ads is provided in advance by the advertiser. The results of this processing (terms, topics), which takes place offline, are indexed in order to be directly accessible without incurring any computational cost. Thus, it is possible for the module to instantly discard from the matching process the subset of ads not belonging to the same topic as the input content item.

For the remaining ad items, the module calculates a graph-based relevance measure with the input content. The recommendation process is complete once the top scoring ad items are selected and displayed to the web surfer that requested the input content item. We introduce two graph-based relevance measures, which, from here on, will be referred to as (a) simple-expansion ($SEXP$) and (b) refined-expansion ($REXP$).

Simple expansion. The simple expansion recommendation technique is carried out in two straightforward steps: (a) deriving an expanded form p_{exp} of the input term vector p and (b) calculating the cosine similarity between the expanded vector and the ad vector α_i: $r_{SEXP} = cos(p_{exp}, \alpha_i)$.

The expanded term vector is derived by collecting the neighbors of the input terms in G_X and then filtering out the ones with lower degree than a given threshold. Once these terms are collected they are merged with the input terms.

Refined expansion. The refined expansion recommendation technique is also executed in two steps: (a) selecting a set of additional terms p_{add} and (b) calculating the relevance measure according to the following equation:

$$r_{REXP} = min(cos(p, \alpha_i) + cos(p_{add}, \alpha_i), 1.0) \quad (4)$$

The additional terms constituting the vector p_{add} are collected by first identifying the most important terms of the input vector p and then using these terms to select the terms of p_{add}. The identification of the important terms in the input vector is based on their degree on the graph in case they are of type NN; in case they are NEs, their co-occurrence (on the graph) with other NEs of the input content is used to derive their significance. Once the selection of important terms takes place, their direct neighbors of the same type (i.e. $NN \rightarrow NN$, $P \rightarrow P$, etc.) on the graph are collected. In that way, assuming that M important terms were identified in the input content, we end up with M term lists of neighbors which are subsequently merged into a single list. Then, the terms of the merged list are re-ordered according to their connections number, i.e. the sum of their co-occurrences with other members of the list. The final expansion vector p_{add} comprises only the top N terms of the re-ordered list.

For both relevance measures presented above, we tested the effect of *keyword boosting*, i.e. boosting of the final relevance score by multiplying the calculated relevance with a boost factor (> 1) in case the graph contained the ad keywords. In that way, we ended up with comparing four different ad-content matching methods, namely $SEXP$, $REXP$ and their *keyword-boosted* variants which from now shall be denoted as $SEXP_{kb}$ and $REXP_{kb}$ respectively.

4 Evaluation

In order to test the potential of the proposed ad recommendation framework, we conducted an experimental study in five topics of commercial interest for online advertising, namely clothing (CLO), soccer (SOC), movies (MOV), music (MUS), and food & restaurants (RES).

4.1 Data Collection and Evaluation Setup

The first data collection step involved the topic graph construction. This was possible by first compiling a list of search keywords related to the five topics of interest and then successively submitting keywords from this list to the Yahoo! Search BOSS service.[3] By collecting the top 50 results from each search, we ended up with five web document collections, based on which we created the respective lexical graphs. On average each corpus contained approximately 1000 web documents.

In the second step, we gathered ad and content data. Since real advertising data is not publicly available, we created a custom ad scraper for the Google

[3] http://developer.yahoo.com/search/boss/

product search service.[4] After manually compiling a set of keywords pertaining to the five topics of interest, we repeatedly submitted them to the service and could finally extract 10,232 ad items of type *Sponsored link* from the search results. We consider these as the pool of ads for our system. The keywords were automatically appended to these ads by putting together the query submitted to the Google product search and the terms that appear in both the ad title and description. Finally, we handpicked 30 web content items per topic (to a total of 150 items) in order to form the input content set for the matching process.

The evaluation process we adopted is similar to the one described in [2]. According to this, we considered the six ad placement strategies described so far, i.e. the two baseline methods (COS and AAK) plus the four graph-based variants ($SEXP$, $SEXP_{kb}$, $REXP$ and $REXP_{kb}$) introduced in this paper. We then selected the top 10 ranked ads provided by each of the six strategies for each of the 150 input content items, resulting in a maximum of 60 ads per content item. These top ads were then inserted in a pool of ads and were submitted to manual evaluation by two independent users. Thus, for each ad set A_i assembled for input item i by the sets $A_{i,X}$ of the six recommenders, a set R_i of relevant ads was identified. Then, assuming that the set of relevant ads contributed by recommender X is $R_{i,X}$, the precision (p) and recall (r), as well as the associated F-measure of X were computed for content item i:

$$p_{i,X} = \frac{|R_{i,X}|}{|A_{i,X}|} \qquad r_{i,X} = \frac{|R_{i,X}|}{|R_i|} \qquad F_{i,X} = \frac{2 \cdot (p_{i,X} \cdot r_{i,X})}{p_{i,X} + r_{i,X}} \tag{5}$$

Since a total of 150 ad recommendations were evaluated by two independent raters, it was possible to derive the statistics (mean, standard deviation) for the performance of each recommender according to each rater and estimate the inter-rater agreement by means of the kappa coefficient introduced in [10].

4.2 Results

Table 1 presents an overview of the performance statistics for the six recommenders under comparison. Inspection of the mean F-measure values attained by the recommenders indicates a clear improvement in the performance of the graph-based recommendation methods over the baseline ones, COS and AAK. The box plot of Figure 3 illustrates this performance improvement. Our confidence in this improvement is further supported by the substantially high agreement between the evaluators, which is inferred from the observed kappa values, $\kappa \in (0.81, 1.0)$. In addition, comparison of the execution time requirements of the different recommenders reveals that the graph-based methods without keyword boosting, i.e. $SEXP$ and $REXP$, are computationally very efficient with respect to the baseline (in fact, $SEXP$ appears to be faster than AAK). [5]

[4] http://www.google.com/products

[5] Obviously, these measurements are meaningful under the assumption that the employed lexical graphs fit into the system memory, which is reasonable since the graphs were created from relatively small corpora.

Table 1. Performance statistics for the six ad recommenders (averaged between the two raters). From left to right: mean precision ($\overline{p}$), recall ($\overline{r}$), F-measure ($\overline{F}$), standard deviation of F-measure ($std(F)$), mean κ coefficient ($\overline{\kappa}$), and execution time ($\overline{t}$).

	$\overline{p}$	$\overline{r}$	$\overline{F}$	$std(F)$	$\overline{\kappa}$	$\overline{t}$ (sec)
COS	0.388	0.292	0.315	0.237	0.863	0.410
AAK	0.445	0.330	0.360	0.242	0.871	0.713
$SEXP$	0.578	0.442	0.471	0.219	0.891	**0.582**
$SEXP_{kb}$	0.594	0.457	**0.487**	0.208	0.824	0.850
$REXP$	0.595	0.436	0.482	0.219	0.830	0.853
$REXP_{kb}$	0.588	0.440	0.480	0.210	0.905	1.120

Finally, by comparing the graph-based recommendation variants to each other, we note that keyword boosting contributes to a marginal improvement in the recommendation relevance. However, this improvement is accompanied by a significant execution overhead; similar overheads are also incurred by the methods based on refined expansion. Thus, we consider $SEXP$ as the preferred recommender for use in practical settings.

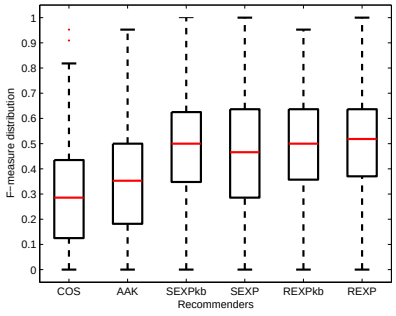

Fig. 3. Box plot of the recommenders F-measure distribution.

Table 2 illustrates in detail how the recommenders perform across the five different topics. The performance of the graph-based recommenders appears to be consistently superior to that of the baseline (except in the case of topic RES where only $SEXP$ and $SEXP_{kb}$ perform marginally better than AAK). Therefore, we may consider that the proposed framework provides improved recommendations across a wide range of input content. This is also confirmed by a series of statistical significance tests on the difference in performance (per topic) between each method and the baseline (COS). For instance, the test results, which are listed in Table 2, indicate that $SEXP$ performs between 11.3% and 19.9% better than COS at a confidence level of $\alpha = 99\%$.

Table 2. Performance of recommenders per topic (average F-measure values) and statistical significance of performance differences. The confidence intervals of the last column are reported at $\alpha = 99\%$ confidence and the interval bounds express % difference from the baseline.

	CLO	**SOC**	**MOV**	**MUS**	**RES**	p-value	Conf. Int.
COS	0.311	0.259	0.247	0.288	0.472	-	-
AAK	0.359	0.317	0.256	0.346	0.524	0.008	0.3 - 8.7
$SEXP$	0.465	0.441	0.405	0.412	0.634	$7.5 \cdot 10^{-5}$	11.3 - 19.9
$SEXP_{kb}$	**0.475**	0.423	**0.454**	0.423	**0.658**	$1.4 \cdot 10^{-4}$	11.6 - 22.7
$REXP$	0.471	0.495	0.402	0.517	0.523	0.007	1.3 - 31.9
$REXP_{kb}$	0.439	**0.507**	0.402	**0.531**	0.522	0.012	-1.0 - 33.6

5 Conclusions

Contextual advertising has emerged as one of the most widely employed forms of online advertising among web content and service providers. While matching web pages to ad items from a large pool of ads, the advertising aggregator needs to maximize the relevance of the recommended ads to the input content item. This task is hindered by the vocabulary impedance problem, which stems from the brevity and idiosyncratic nature of the advertising text.

In this paper, we presented a novel contextual ad recommendation framework based on the notion of lexical graphs. We considered four variants of graph-based ad recommendation and conducted an evaluation study to compare the relevance of the proposed framework recommendations to one offered by the baseline cosine similarity and AAK schemes. We gathered significant evidence that the use of lexical graphs can be beneficial to the task of ad recommendation. Having said that, there is a series of practical issues that one would need to address before applying the proposed framework in a real setting and are thus considered as future work.

First, the web document collection and topic definition is currently carried out in a supervised manner, which renders our approach impractical for web-scale ad recommendation. Integration of topic information from existing large taxonomies, such as Dmoz[6], could alleviate this burden. Further, web-scale application of our framework would create the need for a distributed graph management infrastructure, since as noted earlier, graph-based ad recommendation is computationally efficient under the assumption that the topic graphs can fit into main memory.

Finally, the proposed approach relies to a great extent on the use of NLP tools for the creation of lexical graphs. Therefore, its applicability on text written in languages with little or no NLP support is limited. It is worth investigating whether the sole use of superficial textual features (e.g. term frequency, degree on the graph, etc.) could lead to comparable performance as the current approach but without the need for NLP information (e.g. PoS tags and NE types).

[6] http://www.dmoz.org/

Acknowledgments. We acknowledge support from the MESH and WeKnowIt projects, partially funded by the European Commission, under contract numbers FP6-027685 and FP7-215453 respectively. Furthermore, we thank the anonymous reviewers of the paper for their valuable comments.

References

1. Richardson, M., Dominowska, E., Ragno, R.: Predicting clicks: estimating the click-through rate for new ads. In: Proceedings of the 16th international conference on World Wide Web (WWW '07), pp. 521–530, ACM, New York (2007)
2. Ribeiro-Neto, B., Cristo, M., Golgher, B.G., de Moura, E.S.: Impedance Coupling in Content-targeted Advertising. In: Proceedings of the 28th annual international ACM SIGIR conference (SIGIR '05), pp. 496–503, ACM, New York (2005)
3. Lacerda, A., Cristo, M., Goncalves, M.A., Fan, W., Ziviani, N., Ribeiro-Neto, B.: Learning to advertise. In: Proceedings of the 29th annual international ACM SIGIR conference (SIGIR '06), pp. 549–556, ACM, New York (2006)
4. Murdock, V., Ciaramita, M., Plachouras, V.: A noisy-channel approach to contextual advertising. In: Proceedings of the 1st Workshop on Data Mining and Audience intelligence for Advertising (ADKDD '07), pp. 21–27, ACM, New York (2007)
5. Yih, W., Goodman, J., Carvalho, V.R.: Finding advertising keywords on web pages. In: Proceedings of the 15th international conference on World Wide Web (WWW '06), pp. 213–222, New York (2006)
6. Mishne, G., de Rijke, M.: Language Model Mixtures for Contextual Ad Placement in Personal Blogs. In: Proceedings of 5th International Conference on NLP, FinTAL, pp. 435–446, Springer, (2006)
7. Broder, A., Fontoura, M., Josifovski, V., Riedel, L.: A semantic approach to contextual advertising. In: Proceedings of the 30th annual international ACM SIGIR conference (SIGIR '07), pp. 559–566, ACM, New York (2007)
8. Widdows, D. Dorow, B.: A graph model for unsupervised lexical acquisition. In: Proceedings of the 19th international conference on Computational Linguistics, pp. 1–7, Association for Computational Linguistics, Morristown, USA (2002)
9. Salton, G., Wong, A., Yang, C. S.: A vector space model for automatic indexing. In: Communications of the ACM 18(11), pp. 613–620, ACM, New York (1975)
10. Cohen, J.: A coefficient of agreement for nominal scales. In: Educational and Psychological Measurement 20(1), pp.37–46. (1960)