Inferring Local Synonyms for Improving Keyword Suggestion in an On-line Advertisement System

Luís Sarmento
F. Eng. U. Porto - DEI - LIACC
Rua Dr. Roberto Frias, s/n
4200-465 Porto Portugal
las@fe.up.pt

Paulo Trezentos
ISCTE/ADETTI/Caixa Magica
Av. das Forcas Armadas
1600-082 Lisboa
Paulo.Trezentos@iscte.pt

João Pedro Gonçalves
Portugal Telecom - SAPO
Av. Fontes Pereira de Melo
1069-300 Lisboa, Portugal
joaop@co.sapo.pt

Eugénio Oliveira
F. Eng. U. Porto - DEI - LIACC
Rua Dr. Roberto Frias, s/n
4200-465 Porto Portugal
eco@fe.up.pt

ABSTRACT

In this paper we present a keyword suggestion mechanism for supporting advertisers wishing to publish ads in *content-targeted advertisement systems*. The method infers "synonymy" between keywords by mining a database of previously submitted ads, and uses such information for suggesting *relevant* and *non-obvious keywords* to advertisers. Automatic *word-sense disambiguation* is provided implicitly by our keyword ranking procedure. We perform *on-line* comparison of our method with another keyword suggestion system being currently used in the largest Portuguese web advertisement broker with presence in four different countries, by redirecting 50% of keyword suggestion requests to each of the two systems. We propose a novel set of *evaluation measures* to compare the performance of keyword suggestion systems in such experimental setting. Results show that ads containing keywords suggested by the method we propose are selected for being printed more frequently at similar, or often superior, click-through rates. This results in a potential global revenue increase for the ad broker.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics

General Terms

Algorithms, Experimentation, Measurement, Performance

Keywords

web advertisement, automatic keyword suggestion

ADKDD'09, June 28, 2009, Paris, France.

1. INTRODUCTION

Internet advertisement is one of the main funding sources for many web services, and is the key component for a cost-free Web for the user. There are two types of ads: (i) *display ads*, which are composed by a banner (possibly animated) and are mainly used by larger advertisers, and (ii) *text ads*, which include a *title*, a short *abstract* describing the product or the service being announced, and *URL* pointing to the web site of the advertiser. In this work we will focus on *text ads* because, typically, display ads are not based in keyword-targeting.

When setting a campaign with text ads, advertisers are asked to associate a set of *keywords*[1] that describe the context in which the ad should be placed[2]. *Keyword Targeted Advertisement Systems*, such as Google's AdWords, places ads in the *result page* of web searches. They try to match search terms with the keywords associated with the ads to select the most relevant ones. On the other hand *Content-Targeted Advertisement Systems*, such as Google's AdSense, focus on placing ads on content-rich sites, like newspapers or web logs. These systems perform content analysis in order to extract descriptive terms that will be matched against the keywords ads, so that the most appropriate ads for the content at stake are selected. Thus, advertisers are specially interested in describing their ads using *more* and *better* keywords so they can increase the number of times their ads are printed and clicked by web users, while keeping their overall *cost per click* (CPC) low, and thus getting a better value for their campaigns. However, because of lack of experience or support, in many cases advertisers end up associating only few keywords (1 to 5) to their ads, which are not enough for describing all possible contexts where their ads would be relevant. Thus, many otherwise relevant and highly focused ads are not even considered for placement because

[1]The business terminology for these "keywords" is "*bids*", since they are later involved in a process where ads are chosen to be printed through an *election*.

[2]In fact, keywords (or bids) are usually associated with a *group* of ads that are shown alternatively in order to provide some diversity. For simplicity reasons, throughout this paper we will use the term "ad" for referring to such "group of ads".

the keywords provided by the advertiser are not enough (in number and diversity) to be matched with the appropriate target contexts (either search terms or descriptive terms extracted from content). When no better option exists, brokers place generic ads, but since these are not really targeted for the specific context at stake they have less chances of being clicked by web users. Under a *pay-per-click* scenario, this means less revenue for the broker. In order to reduce this problem, brokers provide *keyword suggestion tools*, such as Google's AdWords Tools[3] or Overture's Keyword Selection Tool[4]. Given an initial set of keywords provided by the advertiser, keyword suggestion tools generate a ranked list of *relevant* and (ideally) *non-obvious* keywords for the advertiser to choose from and associate to its ad, thus enriching the corresponding keyword description.

In this paper, we propose a keyword suggestion mechanism intended for supporting *content-targeted advertisement systems* that mines a database containing previously submitted ads to infer similarity relations among the corresponding associated keywords, and uses such information for suggesting relevant and non-obvious keywords to assist new advertisers. By mining the ads database, we try to identify keywords that can be *interchanged*, i.e. which can be considered *local synonyms* in the universe of known keywords already associated to ads. Suggested keywords are ranked using a function that takes into account both the overlap and the average similarity with keywords provided by the advertiser. The ranking procedure we propose provides an implicit sense-disambiguation, and ensures that suggestions are sense-compatible with the keywords previously given by the advertiser. We perform *on-line experiments* and compare the results of our method with an alternative legacy method. We propose several novel evaluation measures, and we show that our keyword suggestion method outperforms the legacy suggestion method on all these measures, in practically all situations.

2. RELATED WORK

Because of its very high economical impact on web related business, the problem of keyword generation for advertisement has received significant attention lately. Briefly, keyword suggestion tools operate according to few different high-level strategies, sometimes in combination. One approach consists in using information extracted from *web search logs*. Keyword suggestions are those *queries* that lexically include some of the keywords given by the advertiser (e.g. "holidays in Corfu" given "Corfu" or "holidays"). The major limitations of this strategy are (i) the inability of the system to suggest keywords that are relevant but lexically dissimilar, and (ii) the inability to filter out suggestion generated from ambiguous seed keywords. Also, obtaining access to web search logs is not trivial. A second type of approaches consists in using existing lexical-semantic resources, such as thesauri, to perform suggestions. The main limitation of this type of methods is the low recall of the suggestion method since existing resources usually have very low recall and coverage (specially for languages other than English) on many important word classes, such toponyms, names of entities, and words associated with brands or product models (e.g. "Minolta x-700"). A third strategy, which can be applied when the ads database has a large number of ads and enough variety of keywords, consists in mining keyword co-occurrence information from the ads database itself in order to infer relation between keywords chosen by previous advertisers. New keyword suggestion are chosen taking into account such learned keyword relations. This strategy explores the fact that keywords stored the ads database have been subjected to manual selection, and are thus probably relevant for "similar" ads. In this paper, we present a keyword suggestion method that follows such strategy. However, these type of methods are unable to suggest keywords out of the set of those that have already been used (although as the ads database becomes larger and more diverse this might not be such a severe problem). To overcome such limitation, some methods try to infer keyword relations by mining the web. In these cases, possible suggestions are found among the keywords that co-occur with user provided keywords on a set of documents / snippets found on the web (e.g. using a search engine). The main problems of these type of strategies are (i) the complexity of the procedure for extracting keywords from web documents (which can lead to many noisy suggestions), and (ii) the less than ideal response times for real-time processing, since pre-processing is not viable.

Joshi and Motwani introduce TermNet [6], a graph-based technique for identifying semantic relations between keywords using information extracted from search engines. Given a keyword k_i, the method builds a *characteristic document* from text snippets extracted from the top 50 documents found querying a search engine for that keyword. Text snippets are the sentences from those documents containing k_i. Using the corresponding *characteristic documents* it becomes possible to compute the *directed relevance* between pairs of keywords. The authors consider that relevance between two keywords should not be symmetric, in the sense that if k_j is a relevant keyword suggestion for k_i, it does not necessarily mean that k_i is a relevant suggestion for k_j. The directed relevance of k_j to k_i is computed as the frequency of k_j found in the characteristic document of k_i. The resulting directed graph, which expresses the directed relevance between pairs of keywords, is used to perform suggestion. The most interesting keywords for k_i are those whose edges point to k_i. Additional filtering based on *tf-idf* weights [8] can be performed to remove very frequent, and thus less interesting, suggestions. Evaluation is performed by comparing the list of keywords generated for 100 keywords by TermNet and by five other systems. Manual assessors where asked to judge the list for *Relevance* and for *Non-Obviousness* (i.e. suggestion does not contain the initial keyword), and metrics of precision and recall for these indicators were computed. For the top 50 suggestion, precision of TermNet is close to 1 for both *Relevance* and for *Non-Obviousness*. When comparing to other methods TermNet performs better in almost all studied indicators.

Wordy [1] is a similar approach that aims at suggesting keywords that are relevant but at the same time have low frequency and, thus, smaller bidding cost for the advertiser[5].

[3] `https://adwords.google.com/select/KeywordToolExternal`

[4] `http://sem.smallbusiness.yahoo.com/searchenginemarketing/`

[5] This is a rather strong simplification assumption because in practice the relation between frequency and cost of the keyword is not direct. In fact, some low frequency keywords can have very high bidding costs since they are very discrim-

For a given keyword, the idea is to suggest many relevant words with a frequency/cost as low as possible in order to allow the same effect of using high frequency/cost keyword, but with much less global cost for the advertiser. *Wordy* starts by mining the website of the advertiser to find a set of seed keywords based on their *tf-idf* values. These will compose the initial dictionary D_0. In order to expand D_0, a search engine is queried and the top ranked documents retrieved for each keyword in D_0 are added to the corpus. Filtering the corpus by *tf-idf* allows to expands D_0. As in [6], a search engine is queried for each keyword in D_0, and the top ranked documents are used to build the corresponding vector description (again weighted and truncated by *tf-idf*). Pairwise keyword similarity is computed using a kernel equivalent to inner product of the corresponding vectors. The resulting graph is used to perform suggestions: cheaper keywords are found by searching the graph for keywords that are similar and at the same time have lower frequency. The author, however, does not conduct evaluation of the proposed method. In particular, this approach can potentially lead to a decrease in the *conversion rate* (the ratio of users that actually perform the desired action, such as for example purchasing a product), since lower cost keywords may be too far away from the initial context and thus attract users who will find themselves being offered something that they were not expecting.

An alternative approach is proposed in [3]. Instead of using co-occurrence statistics for finding related keywords, the proposed method maps seed keywords onto a concept hierarchy, which is supposed to capture the advertisers goal better than the list of keywords. Once seed keywords have been mapped to the hierarchy, new keywords can be suggested by selecting phrases associated with the closest nodes in the hierarchy. The authors derive the concept hierarchy from the Open Directory Project (ODP), which contains a large set of web pages. First, concepts are derived from the categories of the ODP and are assumed to be organized under a is-a hierarchy. Then, phrases associated with the web document under each category are matched with the corresponding concept node (and with all the parents up the hierarchy). A *tf-idf*-like formula is used to compute the degree of association of phrases with each node, ensuring that phrases that are more peculiar to a given category have a higher weight in the corresponding node. Given a set of seed keywords, the method is able to find the nodes in the hierarchy with higher degree of similarity with them (which may include for example the common ancestor of several matching nodes), and then suggest the corresponding phrases as keywords. When ambiguous seed words are given (i.e. which may be mapped to nodes with distinct senses, having no common ancestor except the top node of the hierarchy) suggestions can be made separately, thus avoiding list of suggestions that mix multiple senses. Authors have shown this property by comparing results of their method against suggestion generated by Google's, Overture's and WordTracker's suggestion tools for the word "matrix". The authors, however, do not address the issues related with the low concept coverage that usually affects concept hierarchies.

Although not explicitly addressing keyword suggestion, the work presented in [2] can be useful while developing keyword suggestion methods. The authors propose a method for clustering the *advertisers* × *keyword* bipartite graph, with the purpose of finding sets of advertisers and sets of keywords that are more strongly connected than the rest of the data set. Such cluster can be seen as *submarkets*, i.e. groups of advertisers that show a common bidding behaviour, and keywords related to the same marketplace. Thus, keywords corresponding to such well defined market places can be suggested to advertisers that have been found to belong to the corresponding group. However, such strong stereotyping will tend to reduce the diversity of keyword suggestion, thus, excluding many otherwise interesting suggestions, specially for those advertisers who do not fit perfectly in a *submarket*. Also, inside each submarket the cost per click for each keyword will tend to increase, since all advertisers will be bidding for approximately the same set of keywords.

inative of a marketing target.

3. METHOD DESCRIPTION

Let $\mathcal{A}$ be the set of advertisements composed of $|\mathcal{A}|$ ads $\mathcal{A} = \{a_1, a_2, ...a_{|\mathcal{A}|}\}$. Each ad can be represented by a tuple of the form $a_i = (t_i, d_i, K_i)$, where t_i is the title, d_i is a short description and K_i is the set of $|K_i|$ keywords provided by the advertiser, $K_i = \{k_{i1}, k_{i2}, ...k_{i|K_i|}\}$. Given a set of seed keywords $K^0 = \{k_1^0, k_2^0, ...\}$ we wish to develop a *keyword suggestion function* $\mathcal{F}_s$ that generates a ranked list of keyword suggestions k_1^s, k_2^s, ... k_n^s that are both *relevant* and *non-obvious* expansions of the initial K^0 set:

$$\mathcal{F}_s(K^0) = \{k_1^s, k_2^s, ...k_n^s\} \quad (1)$$

Let us for now assume that K^{seed} is composed by a single *unambiguous* keyword, i.e. $K^0 = \{k^0\}$. Keyword k^s may be considered a good suggestion for K^{seed} if it is capable of describing the same context described by k_0, or an alternative but *equally relevant* context for the ad at stake. That is, both k_0 and k_s should be comparable in terms of representing equally relevant (but not necessarily the same) contexts for the ad, given the ads domain, $\mathcal{A}$. In other words, if advertisers could *only* associate one keyword from the domain defined by $\mathcal{A}$ to ads, they would chose keyword k^0 or keyword k^s instead with approximately equal probabilities. Keywords k^0 and k^s are, thus, said to be *inter-changeable* in the domain defined by $\mathcal{A}$. From now on, we will use the term *local synonyms* in $\mathcal{A}$, or simply *synonyms*, to refer to keywords that can be inter-changed among ads in $\mathcal{A}$ while maintaining the level of relevancy of the contexts described. We will use the term *synonym* since the behaviour of synonym keywords in the $\mathcal{A}$ domain is similar to the behaviour of synonyms words in common language (i.e. they can be inter-changed without adulterating meaning). Note that in this scope, synonym keywords need not, and probably are not, equivalent to synonyms in the domain of *common language*.

We propose a suggestion method whose definition of relevance is related to the notion of keyword inter-changeability: keyword s is considered a relevant suggestion for an initial seed set $K^0 = \{k_1^0, k_2^0, ...\}$ set if it is inter-changeable with one or more elements from K^0. Additionally, the larger the number of elements of K^0 with which k^s can be swapped, the more relevant will k^s tend to be as a suggestion for K^0. In fact, as we will show later, when ambiguous keywords exist in initial the seed set the level of inter-changeability can be used for suggesting correct keywords.

3.1 Computing Keyword Synonymy

Two words can be considered synonyms in the domain of keywords associated with ads set $\mathcal{A}$, if they systematically co-occur with the same *previously known* keywords (i.e. with those that have already been selected by advertisers up to that point). The intuition here is that, if two keywords k_i and k_j are in fact inter-changeable, advertisers will associate either one of them to an ad. Thus, assuming that there are ads concerning the same range of product, in some cases advertisers will choose to associate k_i while in other similar ads will associate k_j, while keeping the rest of relevant keywords. Let us assume that we have a dataset composed of a large number of ads, collected over the time. We can compile statistics regarding the co-occurrence of keywords in the sets of keywords associate with each ad, K_j. Let $[C(k_i)]$ be the vector of keyword co-occurrences for keyword k_i, which contains information about the number of ads (or keyword sets K_j) in which keyword k_i co-occurs with all other keywords:

$$[C(k_i)] = [(k_1, f_{i1}), (k_2, f_{i2}), ...(k_i, 0)...(k_n, f_{in})] \quad (2)$$

Let us also assume that there is a *feature weighting function* $\mathcal{W}$ that is used to ponder features by computing the degree of association between the co-occurring keywords. $\mathcal{W}$ will allow to reduce the relative importance of the co-occurrence of k_i with very frequent, and thus less strongly related, keywords. Let $[C_{\mathcal{W}}(k_i)]$ be the vector that results from applying the feature weighting function to $[C(k_i)]$:

$$[C_{\mathcal{W}}(k_i)] = \mathcal{W}([C(k_i)]) \quad (3)$$

Common weighting functions include *tf-idf* [8], *Mutual Information* [4] and the *Log-Likelihood Ratio* [5]. The degree of synonymy s_{ij} between two keywords, k_i and k_j can now be computed by applying a generic vector similarity metric $\mathcal{S}$ to the two corresponding feature-weighted co-occurrence vectors:

$$s_{ij} = \mathcal{S}\left([C_{\mathcal{W}}(k_i)], [C_{\mathcal{W}}(k_j)]\right) \quad (4)$$

Possible instantiations of $\mathcal{S}$ include the *cosine metric* [8] and the *Jensen-Shannon Distance* [7]. A keyword synonymy graph, $\mathcal{G}_s$, can be obtained by computing pairwise similarity between all co-occurrence vectors. For some keywords pairs, s_{ij} will be null or very low. Filtering by s_{ij} will allow to keep only the strongest synonym links and remove noisy edges. To allow a more intuitive visual understanding of the graph, s_{ij} weights (corresponding to strength of the synonymy relation) can be substituted by distance weights, which vary inversely with the strength of the synonymy. Thus, nodes with high degree of synonymy will be located closer to each other than nodes with a lower degree of synonymy.

3.2 Keyword Suggestion and Ranking

The similarity graph $\mathcal{G}_s$ can be used for keyword suggestion in a straight-forward way. Given a seed keyword, k_1^0, relevant keyword suggestions may be found amongst the nodes of $\mathcal{G}_s$ closest to the k_1^0 node, i.e. those corresponding to keywords with highest synonymy values. Suggestions can be ranked by the length of edges to the seed node: the best keyword suggestions will be located closer to the seed node. Looking at Figure 1 this corresponds to saying that $\mathcal{F}_s(k_1^0) = \{k_A^s, k_B^s, k_C^s, k_D^s\}$, ranked by the order shown.

If more than one seed keyword is given then more information describing the goal of the advertiser is available, which should be used for improving keyword suggestions. Obviously, extra seed keywords allow finding *more* relevant keywords since the number of nodes around seed keywords is expected to grow. More important, with more keywords there is the chance of resolving possible ambiguity related with some ambiguous seed keywords, and suggest only keywords that are related to the sense that is relevant to the advertiser.

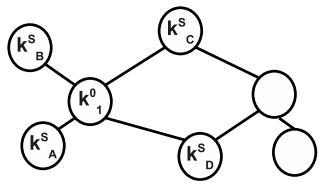

Figure 1: Keyword selection and ranking for one seed keyword.

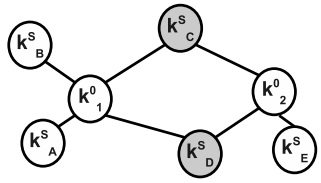

Figure 2: Keyword selection and ranking for two seed keywords.

Consider for example that seed keyword k_1^0 from example given in Figure 1 is ambiguous (e.g. k_1^0 = "orange"). It is possible that the set of four keywords suggested may correspond to relevant keywords, but only relevant to one of the different senses of k_1^0 (e.g. k_A^s = "apple", k_B^s = "banana", k_C^s = "yellow" and k_D^s = "red"). If an additional seed keyword is known, k_2^0, then it might be possible to infer which sense of k_1^0 is relevant for the advertiser and exclude incorrect suggestions that are derived from other senses. This can be done by considering that suggestion that correspond to the correct sense of k_1^0 will probably also be synonyms of k_2^0. By intersecting the sets of synonyms of k_1^0 and of k_2^0 we will find such keywords, even if they are not the ones that, individually, have the highest level of synonymy with each of the seed keywords. Figure 2 illustrates on such situation (assume for example that k_2^0 = "blue" and k_E^s = "green"), where keywords, k_C^s and k_D^s, despite not being the ones that have higher synonymy with any of the seed keywords, are the ones that are synonym with both k_1^0 and of k_2^0, and should thus be ranked higher than k_A^s, k_B^s and k_E^s.

3.3 Overview of the Suggestion Procedure

In summary, given a set of seed keywords, $K^0 = \{k_1^0, k_2^0, ...\}$, with one or more keyword seeds k_i^0, and the synonymy graph, $\mathcal{G}_s$, our suggestion algorithm works as follows:

1. for each $k_i^0 \in K^0$ obtain the set of direct neighbours in $\mathcal{G}_s$, $K_i^n = \{(k_{i1}^n, s_{i1}), (k_{i2}^n, s_{i2})...\}$, where s_{ij} represents the degree of synonym between k_i^0 and k_{ij}^n as taken from $\mathcal{G}_s$;

2. merge all sets of neighbours K_i^n in a single set K^N, while computing for each keyword $k_j^N \in K^N$:

(a) *keyword overlap*, o_j, as the number seed keywords k_i^0 to which k_j^N is a direct neighbour (i.e. for which k_j^N is found in the corresponding neighbour set, K_i^n);

(b) *average degree of synonymy*, $\overline{s}_j$, as the average degree of synonym that was found between keyword k_j^N and each of the seed keywords to which k_j^N is direct neighbour.

to obtain $K^N = \{(k_1^N, o_1, \overline{s}_1), (k_2^N, o_2, \overline{s}_2)...\}$;

3. Rank K^N first by descending values of o_i and then by descending values of s_i. Keyword suggestion, k_s are taken from the top ranked keywords in K^N.

Therefore, keywords that are synonyms of many seed keywords always rank higher than those that are synonyms of only a few seed keywords. As explained before, this provides implicit filtering against irrelevant suggestion generated from ambiguous seed keywords.

One possible criticism regarding our method is that it might introduce certain distortions in the bidding process, since it suggest keywords submitted by *previous* advertisers to help new advertisers. In a competitive environment, as advertisement bidding is, it would be unfair if new advertisers could benefit from the good keywords ideas that were submitted by a previous advertisers, who would then have additional competitors in the bidding process for such keywords.

However, our ranking method indirectly prevents that from happening. As explained before, suggested keywords that overlap with several input keywords are ranked higher than those that overlap with only a few keywords. This means that, in practice, keywords that have already been submitted by more advertisers, and thus co-occur with more keywords, are placed higher in the ranked lists of suggestions (suggestions ranked lower are simply filtered out). Thus, our suggestion system tends to suggest keywords that have already been submitted by more advertisers, while keeping individual "business secrets" hidden. In certain way, what our keyword suggestion method provides is the "wisdom of the crowd", specially to the less experienced advertisers.

In any case, advertisers that rely *exclusively* on the suggestions proposed, are always more exposed to competitive bidding. Ideally, advertisers should apply a strategy that combines automatically suggested keywords, which represent mainstream traffic, with more "exclusive" keywords, which are less exposed to competition.

4. ON-LINE EVALUATION OF KEYWORD SUGGESTION SYSTEM

Let us start by defining *campaign publishing session* as the complete set of steps that an advertiser has to follow to submit one advertisement to the web advertisement system. A *campaign publishing session* includes (i) choosing the name of the campaign, (ii) choosing the content of the ad and (iii) choosing the keywords for the ad, either by typing them in directly, or by selecting keywords provided by the suggestion system. Obviously, the suggestion systems require at least one initial seed keyword provided by the advertiser. Once a list of suggestions has been generated, the advertiser can either pick keywords from a that list or type in new keywords. The advertiser can then request new keywords suggestion, which will be generated using the previously added keywords as new seeds. The process continues iteratively until the advertiser is happy with the set of keywords chosen and performs final submission of the ad.

Performing evaluation of keyword suggestion systems is not trivial, because it is an highly application-oriented task, and because no well established standards exist. Therefore, evaluation can focus on two possible perspectives: (i) semantic (i.e. relevancy) or formal (e.g. "non-obviousness") criteria, and (ii) user feedback (i.e. is the system useful for advertiser?). Most works so far have performed evaluation by manually testing a small sample of suggested keywords to check if they are relevant and non-obvious. When there are ambiguous keywords in the seed sets, tests focus on checking if the system is able to either separate the suggestion by possible senses, or automatically infer the sense that best expresses the purpose of the advertiser. Frequently, authors manually compare keyword suggestions generated by their methods against those generated by Google's AdWord's or Overture's suggestion tools. Instead, we will perform *on-line* evaluation of our method, and compare it against a legacy keyword suggestion system that has been providing suggestions for the `http://anuncios.sapo.pt` web advertisement platform for about 5 years (since March 2004). Evaluation will be based on statistics gathered directly from the behaviour of advertisers. As far as we know, this is the first time that evaluation of keyword suggestion systems for web advertisement platforms based on real usage is reported.

We wish to evaluate our method and compare it with the legacy suggestion system. Moreover, both systems will be assessed regarding the *usefulness* of the method for the advertisers, and also how much impact does the method have in the revenue it is able to generate. More specifically:

- Method Usefulness - Do advertisers choose keywords suggested by the new method frequently? Do advertisers accept suggestions ranked high by the new system, or do they choose suggestion ranked lower? What is the ratio of the number keywords directly typed-in by the advertiser vs. the number of automatically suggested keywords?

- Impact on Revenue - Are ads which received keywords suggested by the new system selected for being *printed* on web pages more frequently than the ones which received keyword suggestions generated by the legacy system? Do they end up being *clicked* more often, and hence generate more revenue for the broker?

Given a legacy keyword suggestion function, $\mathcal{F}_L$, and a new keyword suggestion function $\mathcal{F}_N$, one can perform comparative on-line evaluation by having both systems running in parallel and randomly choosing one or the other to assist the advertiser in a given *campaign publishing session*. This means that the keyword suggestion function is chosen for the *entire* session: all keywords suggested in that session are generated by one and only one of the two competing systems. Let $\mathcal{F}_N$ be activated with probability p_N, and $\mathcal{F}_L$ be activated with probability $p_L = 1 - p_N$. For each new ad, a_i, submitted to the system we keep information about which function used to generate suggestions (either $\mathcal{F}_N$ or $\mathcal{F}_L$). Then, for each individual keyword that the advertiser chooses to associate with the ad, we log if it was directly typed in by the advertiser, or if it was selected from the set

of keywords generated by the suggestion system that was previously activated for that specific session.

4.1 Evaluating Method Usefulness

For each *campaign publishing session* the following information is logged for all keywords, k_{ij}, associated with ads submitted by advertisers, a_i:

- *source*, $\int(k_{ij})$, whether the keyword was directly typed in by the user ($\int(k_{ij}) = U$), or was suggested by one of the two available suggestion functions, the *legacy* suggestion function, $\mathcal{F}_L$ ($\int(k_{ij}) = L$)), or the *new* suggestion function $\mathcal{F}_N$ ($\int(k_{ij}) = N$).
- *rank*, $r(k_{ij})$: the position at which the keyword was ranked in the list of suggestions, when suggested by $\mathcal{F}_L$ or $\mathcal{F}_N$.
- *iterations until selection*, $i(k_{ij})$: the number of the suggestion iterations used by the advertiser until the keyword was added to the ad.

Based on these statistics we can compute several indicators for each ad. The first is the ratio between the number of automatically suggested keywords and the total number of keyword associated with the ad, including those directly typed in by the user. Let $\int_{user}(i)$ be the number of keywords associated with ad a_i directly typed in by the user, and $\int_{auto}(i)$ be the number of keywords suggested by either $\mathcal{F}_L$ or $\mathcal{F}_N$ (only one of the suggestion function is used for each ad). Then, the *automatic suggestion ratio* for ad a_i is given by:

$$\mathcal{S}_r(i) = \frac{\int_{auto}(i)}{\int_{user}(i) + \int_{auto}(i)} \quad (5)$$

Values of $\mathcal{S}_r(i)$ will range from 0 to 1. Values higher than 0.5 mean that users are accepting more keywords suggested by the automatic suggestion system than the one that they are typing in. For each suggestion function, a global performance figure, $\overline{\mathcal{S}_r}$, can be obtained by averaging $\mathcal{S}_r(i)$ over all ads with keywords suggested by that function. The suggestion function – $\mathcal{F}_L$ or $\mathcal{F}_N$ – that has higher $\overline{\mathcal{S}_r}$ can be seen as more *useful* for the advertiser.

Data regarding the position at which the suggested keyword was ranked in the list of suggestions, $r(k_{ij})$, will help to test if ranking procedure is compatible with implicit preferences of the advertiser. If ranking procedure is efficient, top ranked keyword suggestions should be chosen more frequently than the others. For each ad a_i we can compute:

- *average suggestion rank*, $\mathcal{R}(i)$: the average of rank of the suggested keywords selected, $r(k_{ij})$;
- $\mathcal{T}_{@1}(i)$: the fraction of suggested keywords at rank 1 selected by the user;
- $\mathcal{T}_{@10}(i)$: the fraction of suggested keywords up to rank 10 selected by the user;

Again, global performance figures can be obtained for each suggestion function by averaging the previous statistics, $\mathcal{R}(i)$, $\mathcal{T}_{@1}(i)$ and $\mathcal{T}_{@10}(i)$, over all ads with keywords suggested by that system, to obtain $\overline{\mathcal{R}}$, $\overline{\mathcal{T}_{@1}}$ and $\overline{\mathcal{T}}_{@10}$, respectively.

Finally, the number of *iterations until selection*, $i(k_{ij})$, will allow us evaluate the period during which the suggestion system is capable of presenting *novel* useful keywords that are still relevant to the advertiser. For each ad a_i we can compute $\mathcal{I}(i)$, the average on number of the iterations at which the suggested keywords were chosen (always larger that one). A global performance value for each of the suggestion function can again be computed by averaging $\mathcal{I}(i)$ over all ads with keywords suggested by the system at stake to obtain $\overline{\mathcal{I}}$.

4.2 Impact on Revenue

We also wish to compare the impact of using the new suggestion function $\mathcal{F}_N$ on indicators associated with the revenue that ads are capable of generating. We have the following run-time information available for all keywords, k_{ij}, associated with an ad a_i:

- $\#_{imp}(i,j)$: the number of times that ad a_i was selected for being printed in a web page as a result of the bid associated with keyword $k_{i,j}$;
- $\#_{clk}(i,j)$: the number of clicks on a_i after being printed on a web page as a results of a bid on keyword $k_{i,j}$

For each suggestion function we can compute the following set of statistics regarding the generated keywords:

- *average keyword printability*, $\overline{\mathcal{P}_k}$, the average number of ad prints that were made as result of a bid placed on a suggested keyword;
- *average keyword clickability*, $\overline{\mathcal{C}_k}$, the average number of clicks made on ads that were printed as result of a bid placed on a suggested keyword;
- *keyword printabilty efficiency*, $\epsilon_k^{\mathcal{P}}$, the fraction of suggested keywords that lead the corresponding to ad being printed;
- *keyword clickability efficiency*, $\epsilon_k^{\mathcal{C}}$, the fraction of suggested keywords that lead the corresponding ad being clicked;

We can also compute statistics that reflect the overall impact of the suggestion systems on the *ads*. Based on counts $\#_{imp}(i,j)$ and $\#_{clk}(i,j)$ regarding automatically suggested keywords *only*, we can compute:

- *average ad printability*, $\overline{\mathcal{P}_a}$: the average number of times an ad is printed as result of an automatically suggested keyword;
- *average ad clickability*, $\overline{\mathcal{C}_a}$: average number of times an ad is clicked as a result of an automatically suggested tag;
- *average click through rate*, $\overline{CTR}$: ratio between $\overline{\mathcal{C}_a}$ and $\overline{\mathcal{P}_a}$

From all these statistics, clickability related ones – $\overline{\mathcal{C}_k}$, $\epsilon_k^{\mathcal{C}}$ and specially $\overline{\mathcal{C}_a}$ – are the ones that best reflect the impact of the suggestion system on the revenues of the broker. However, printability statistics – $\overline{\mathcal{P}_k}$, $\epsilon_k^{\mathcal{P}}$ and specially $\overline{\mathcal{P}_a}$ – are also extremely important because they reflect whether suggested keywords help the broker in choosing content-targeted ads instead of printing default ads, which are content-agnostic.

5. EXPERIMENTAL SET-UP

For computing keyword synonymy by the process described in Section 3.1, we used a set of 84,180 ads, compiled over a period of about 5 years, during which the web interface used by advertisers only had the legacy suggestion function available. In average, these ads have 14.14 keywords, but 63% only have one keyword associated and almost 70% have 5 or less keywords associated. Ads with more that 75 keywords (2022) were ignored to avoid *catch-all* ads that have long lists of keywords, most of them semantically unrelated. Ads with duplicate set of keywords (8434) were also ignored since we considered that they do not bring useful synonym information. For the set of valid ads, we compiled all keyword co-occurrence pairs. We obtained co-occurrence information for a set of 122,099 keywords.

Each of these keywords was represented by a feature vector whose components are the values of the co-occurrence with other keywords (see Equation 2). Vectors components were then weighted by Mutual Information [4] (see also Eq. 3):

$$MI(k_m, k_n) = \log_2 \frac{P(k_m, k_n)}{P(k_m) \cdot P(k_n)} \quad (6)$$

where $P(k_m, k_n)$ represents the probability of co-occurrence between keywords k_m and k_n, in $\mathcal{A}$ (the set of ads) and $P(k_m)$ and $P(k_m)$ represent the probability of occurrence of each keyword. Next, all-against-all vector comparison using the cosine metric was performed in order to obtain the keyword synonymy graph, $\mathcal{G}_s$ (see also Equation 4):

$$cos\left([C_{\mathcal{W}}(k_i)], [C_{\mathcal{W}}(k_j)]\right) = \frac{[C_{\mathcal{W}}(k_i)] \cdot [C_{\mathcal{W}}(k_i)]}{||C_{\mathcal{W}}(k_j)|| \times ||C_{\mathcal{W}}(k_j)||} \quad (7)$$

For each node (i.e. keyword) we kept only the top 100 closest nodes (i.e. most similar keywords), in order to simplify the link graph.

Given the synonymy graph, $\mathcal{G}_s$, and the ranking procedure described in Section 3.2 we set up the new keyword suggested function, $\mathcal{F}_N$ running side by side with the existing legacy function $\mathcal{F}_L$ in the web platform dedicated to advertisers. The legacy function $\mathcal{F}_L$ combines three methods for suggesting keywords from the set keywords directly typed in by the advertiser:

1. using the OpenOffice thesaurus for Portuguese[6] to find related words;
2. selecting from the query logs of a commercial web search engine the most frequent search queries that lexically include the user defined keywords (this tends to generate a very high number of suggestions);
3. select those keywords from the ads already in ads database that lexically include user defined keywords.

When starting a new campaign publishing session, one (and only one) of the two suggestion function, either $\mathcal{F}_N$ or $\mathcal{F}_L$, is activated with 50% probability. The activated suggestion function is then used throughout the *entire* publishing session, so that *all* keywords suggested for a given ad come either from $\mathcal{F}_N$ or from $\mathcal{F}_L$. We kept this configuration running for a period of 15 weeks. Unfortunately, most of advertisers already have predefined lists of keywords for associating with their ads, so only a small subset of them (approx. 5%) actually uses automatic keyword suggestion. Therefore, we only compared 192 ads for which at least one keyword was suggested by either one of the two suggestion function available: 69 ads have keywords suggested by $\mathcal{F}_N$ and 132 have keywords suggested by $\mathcal{F}_L$.

[6]Available through `ptopenthesaurus.caixamagica.pt`

6. RESULTS AND ANALYSIS

We will present statistics considering two possible analysis scenarios: (1) including *all* ads, even those that can be considered *catch-all ads*[7] ($\mathcal{F}_N^{all}$ and $\mathcal{F}_L^{all}$), and (2) by including only ads with no more than 75 keywords ($\mathcal{F}_N^{75}$ and $\mathcal{F}_L^{75}$). Table 1 presents some statistics about the ads analyzed in both scenarios, for each suggestion function. We present (i) the number of ads that use each suggestion function, $\#_{ads}$, (ii) the average and median number of keywords associated with each ad, $\overline{\#_{kwrd}}$ and $\widehat{\#_{kwrd}}$, and (iii) the average and median number of keywords *automatically suggested* by the active suggestion system, $\overline{\#_{kwrd}^{sug}}$ and $\widehat{\#_{kwrd}^{sug}}$. Scenario 1 includes about 20-35% more ads than Scenario 2, i.e. it includes those ads that can be considered *catch-all* ads. Also, in both scenarios advertisers tend to select more keywords suggested by $\mathcal{F}_N$ than by $\mathcal{F}_L$, even in Scenario 2 in which the average number of keywords per ad is similar. However, there are less ads with keywords suggested by $\mathcal{F}_N$ than there are ads with keywords by $\mathcal{F}_L$. One possible explanation for this is inability of $\mathcal{F}_N$ to suggest keywords as a response to "unknown" keywords input by the advertisers (i.e. to keywords that have not been found in ads used for generating the link graph). If the first and only keyword directly provided by the advertiser is one of such unknown keywords, no suggestions can be made, which will leave the advertiser frustrated, making them possibly quit the session. When restarting the procedure later, there are chances of $\mathcal{F}_L$ being chosen as the active suggestion function, which does not suffer from this problem so severely.

	Scenario 1		Scenario 2	
	$\mathcal{F}_L^{all}$	$\mathcal{F}_N^{all}$	$\mathcal{F}_L^{75}$	$\mathcal{F}_N^{75}$
$\#_{ads}$	123	69	103	51
$\overline{\#_{kwrd}}$	51.3	93.2	27.7	27.8
$\widehat{\#_{kwrd}}$	6	11	4	8
$\overline{\#_{kwrd}^{sug}}$	19.1	32.8	7.5	11.3
$\widehat{\#_{kwrd}^{sug}}$	3	8	2	6

Table 1: Statistics about the ads compared under both scenarios.

6.1 Method Usefulness

Table 2 shows statistics related with how advertisers use suggestion functions. Specifically, it presents values regarding suggestion ratio, $\overline{\mathcal{S}_r}$, average suggestion rank, $\overline{\mathcal{R}}$, the ratio of keywords at rank 1 and up to rank 10 selected by the user, $\overline{\mathcal{T}_{@1}}$ and $\overline{\mathcal{T}_{@10}}$, and the average number of the iteration at which suggested keywords were selected, $\overline{\mathcal{I}}$.

[7] *Catch-all ads are those ads to which advertisers have more or less indiscriminately associated hundreds or thousands of keywords in order to cover a wider range of possible contexts.*

	Scenario 1		Scenario 2	
	$\mathcal{F}_L^{all}$	$\mathcal{F}_N^{all}$	$\mathcal{F}_L^{75}$	$\mathcal{F}_N^{75}$
$\overline{\mathcal{S}_r}$	**0.37**	0.35	0.27	**0.40**
$\overline{\mathcal{R}}$	154.9	**129.1**	65.1	**27.6**
$\overline{\mathcal{T}_{@1}}$	0.09	**0.14**	0.06	**0.32**
$\overline{\mathcal{T}}_{@10}$	0.17	**0.20**	0.22	**0.49**
$\overline{\mathcal{I}}$	1.04	**1.32**	1.14	**1.34**

Table 2: Statistic regarding advertiser's use of the suggestion function.

There are significant differences between both scenarios, which confirms that there are in fact good reasons for considering these two separate scenarios. Except for one case, $\mathcal{F}_N$ scores better in all indicators. Statistics $\overline{\mathcal{R}}$, $\overline{\mathcal{T}_{@1}}$ and $\overline{\mathcal{T}_{@10}}$ show that advertisers tend to pick suggestions ranked higher when using $\mathcal{F}_N$ than when using $\mathcal{F}_L$. This is specially so in Scenario 2, where comparison is not biased by the very long lists of keywords associated with catch-all ads. The higher value of $\overline{\mathcal{I}}$ shows that advertisers requests suggestion more often when using $\mathcal{F}_N$. Statistic $\overline{\mathcal{S}_r}$ relates the number of suggested keywords selected by the user with the one he/she directly types and show a different behaviour. In Scenario 1, both $\mathcal{F}_N$ and $\mathcal{F}_L$ have very similar and relatively high values for $\overline{\mathcal{S}_r}$, which is not surprising since advertisers wishing to create *catch-all* ads will tend to pick as many words as possible more or less indiscriminately. However, in Scenario 2, where advertisers are expected to be more selective, one can see that $\overline{\mathcal{S}_r}$ increases for $\mathcal{F}_N$ and drops significantly for $\mathcal{F}_N$. The average number of iterations, $\overline{\mathcal{I}}$, is always higher for $\mathcal{F}_N$, but this is probably related to the fact that the first iteration of $\mathcal{F}_N$ tends to generate a small number of suggestions if the advertiser has only given one seed keyword. Therefore, in such conditions advertisers will almost always have to request additional suggestion iterations.

6.2 Impact on Revenue

Table 3 presents statistics about the contribution of suggested keywords to the number of times ads are printed and clicked (i.e. all these statistics take into account *only* data coming from suggested keywords).

	Scenario 1		Scenario 2	
	$\mathcal{F}_L^{all}$	$\mathcal{F}_N^{all}$	$\mathcal{F}_L^{75}$	$\mathcal{F}_N^{75}$
$\overline{\mathcal{P}_k}$	6865.4	**13896**	1065.4	**2679.0**
$\overline{\mathcal{C}_k}$	**10.0**	9.7	0.80	**2.35**
$\epsilon_k^{\mathcal{P}}$	0.17	**0.18**	0.11	**0.22**
$\epsilon_k^{\mathcal{C}}$	0.071	**0.08**	0.048	**0.10**

Table 3: Printability and clickability statistics per keyword suggested

There are several significant facts in these statistics. First, there is a very large difference in the number of average prints and clicks generated by suggested keywords between both scenarios. This basically suggests that there are a few ads in Scenario 1 that generate an enormous amount of prints and clicks, and that are not included in Scenario 2. We manually verified the presence of a few of such "outliers". The second interesting fact is that the number of prints generated by keywords suggested by $\mathcal{F}_N$ is always much higher. For Scenario 2, such difference is more than 150%. As for the average number of clicks per ad due to suggested keywords, $\overline{\mathcal{C}_k}$, the situation is a bit different. While in Scenario 1, $\overline{\mathcal{C}_k}$, is quite similar for both $\mathcal{F}_N$ and $\mathcal{F}_L$, in Scenario 2 $\mathcal{F}_N$ scores almost three times as much. If we look at keyword printabilty and clickability efficiencies (i.e. the fraction of suggested keywords that actually generate a print and then a click), $\epsilon_k^{\mathcal{P}}$ and $\epsilon_k^{\mathcal{C}}$, in all cases keywords suggested by $\mathcal{F}_N$ score higher. Again, in Scenario 2, $\mathcal{F}_N$, outperform keywords suggested by $\mathcal{F}_L$ at least 100%.

Results presented in Table 4 summarize the impact of the suggested keywords on the overall printability and clickability of ads. Statistics $\overline{\mathcal{P}_a}$ and $\overline{\mathcal{C}_a}$ indicate, respectively, the average number of times that ads are printed and clicked due to a bid on any of their suggested keywords.

	Scenario 1		Scenario 2	
	$\mathcal{F}_L^{all}$	$\mathcal{F}_N^{all}$	$\mathcal{F}_L^{75}$	$\mathcal{F}_N^{75}$
$\overline{\mathcal{P}_a}$	131,279	**450,713**	7,995	**30,309**
$\overline{\mathcal{C}_a}$	191.5	**312.1**	5.99	**26.59**
$\overline{CTR}$ ($\times 10^{-3}$)	**1.45**	0.69	0.74	**0.87**

Table 4: Printability and clickability statistics for ads based on suggested keywords only.

Again, in both scenarios these printability and clickability statistics, $\overline{\mathcal{P}_a}$ and $\overline{\mathcal{C}_a}$, are higher for ads with suggestions from $\mathcal{F}_N$. In Scenario 2, ads with suggestions from $\mathcal{F}_N$ are printed and clicked approximately *4 times more* then ads with keywords suggestions coming from $\mathcal{F}_L$. The click-through rate is also about 17% higher for $\mathcal{F}_N$ ads. However, although in Scenario 1 ads with keywords from $\mathcal{F}_N$ do generate more prints and clicks than those with keywords from $\mathcal{F}_L$, the click-through rate of the last is much higher than of the first. The reason for this is related to a small set of atypical ads. Figure 3 compares click-through rates (CTR) at several values for the threshold on the maximum number of keywords for an ad to be considered valid. It shows that CTR values are higher for ads with keywords from $\mathcal{F}_N$ when the threshold is lower than 100 keywords, except for one case (threshold=50). It also clearly shows a peak for $\mathcal{F}_L$ when threshold is set 200, as a result of a few outliers found in that range. Given these results, and excluding the influence of the outliers which we believe would be much less significant if more data was available, it is quite reasonable to say that $\mathcal{F}_N$ can almost always lead to ads with better CTR values than those obtained by ads with keywords suggested by $\mathcal{F}_L$. In any case, from an absolute point of view, keywords from $\mathcal{F}_N$ generate more prints and clicks in both scenarios.

We will now analyze these values taking into account statistics concerning the globality of ads submitted by advertisers during the test period. We will only analyze ads with no more than 75 words. At this threshold, there are 2684 ads with no keywords automatically suggested by any of the two systems. As it is possible to confirm, only a small fraction of ads – 5.42% – have automatically suggested keywords. Notably, ads with no suggested keywords have much inferior av-

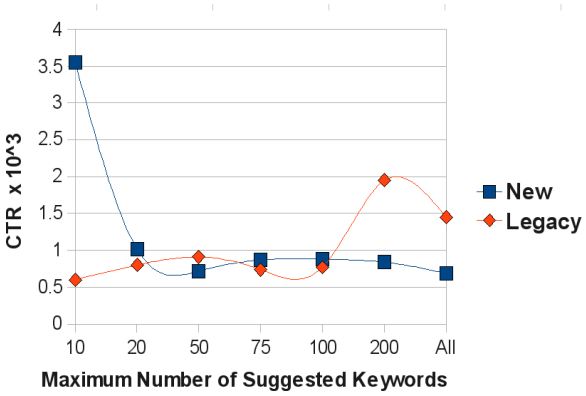

Figure 3: CTR for ads with keyword suggested by $\mathcal{F}_N$ and $\mathcal{F}_L$, at several thresholds on the maximum number of keywords.

erage CTR values (0.15×10^{-3}), which shows the usefulness of keyword suggestion systems. During the analysis period, ads with no suggested keywords were printed 3,591,028,887 times and generated 551,864 clicks. We will start by assuming that only the new suggestion method was available and that all the 154 ads with suggested keywords had received the suggestions from the new suggestion method. We can thus admit that they would all have the same average printability and clickability statistics per ad that are seen in Table 4 for $\mathcal{F}_N$ (i.e. $\overline{\mathcal{P}_a} = 30,309$ and $\overline{\mathcal{C}_a} = 26.59$). In such case, these ads would generate 2,298,301 additional prints and 2,120 more clicks. Globally, this represents relative increases of only 0.06% prints but a 0.38% increase in clicks. Although modest at this stage, these numbers could become much more important when the fraction of advertisers using the keyword suggestion mechanism become higher (currently only around 5.0%).

7. CONCLUSION AND FUTURE WORK

We presented a keyword suggestion mechanism that mines information from a database of previously known ads in order to infer *local synonymy* information between keywords. The method exploits the fact that keywords previously assigned to ads have already gone through a relevancy selection procedure made by previous advertisers, and uses synonymy information to perform *relevant* (and *non-obvious*) suggestions, while automatically performing implicit sense-disambiguation. We performed *on-line* evaluation of our system by comparing it against a legacy keyword suggestion system. Both systems were exposed to real advertisers during 15 weeks. As far as we know, this is the first study of this type that is reported. We proposed a set of novel performance measures for such an experimental setting. Using these measures we showed that keywords suggested by our system outperform keywords suggested by the legacy system in several parameters related to printability and clickability. Moreover, we showed that ads with keywords suggested by our system are printed more often and clicked more frequently than those with keywords suggested by the legacy system, and that they also tend to have higher CTR values.

Future work will involve trying to solve the main limitations of our suggestion mechanism. There are two key points we should address. The first concerns the inability to suggest keywords as a response to "unknown" keywords input by the advertisers. We believe that this has severely reduced the number of times the system was effectively used. The second, which is somehow related, is the inability of our system to generate suggestions out of the set of known words. Obviously, as the number of the ads that we have on the database increases, the chances that such situations occur decreases, but there is additional room for improvement. One possible solution for both these problems is to mine keyword co-occurrence information from other media that we have available, namely *search query logs* and *blog content*. This will significantly increase the number of known keywords, from a large variety of domains, and such information can be used either as a complement to the information mined from ad logs, or simply as a backup mechanism. Although currently the global impact of our method on the global revenue generated for the ad broker is not significant, we believe that, as advertisers become more aware of the importance of the keyword suggestion mechanisms in the effectiveness of their campaigns, the positive impact on broker's revenue of our suggestion method will become more relevant.

Acknowledgments

The authors wish to thank the Sapo.pt Ads team, and more specifically to Francisco Temudo, for the help provided in setting up the experimental framework over the legacy keyword suggestion system, and for their suggestions throughout the preparation of this work. This work was partially supported by grant SFRH/BD/23590/2005 FCT-Portugal, co-financed by POSI.

8. REFERENCES

[1] V. Abhishek and K. Hosanagar. Keyword generation for search engine advertising using semantic similarity between terms. In *ICEC '07: Proceedings of the ninth international conference on Electronic commerce*, pages 89–94, New York, NY, USA, 2007. ACM.

[2] J. Carrasco, D. Fain, K. Lang, and L. Zhukov. Clustering of bipartite advertiser-keyword graph. In *Proc. International Conference on Data Mining (ICDM'03)*, Melbourne, Florida, November 2003.

[3] Y. Chen, G.-R. Xue, and Y. Yu. Advertising keyword suggestion based on concept hierarchy. In *WSDM '08: Proceedings of the international conference on Web search and web data mining*, pages 251–260, New York, NY, USA, 2008. ACM.

[4] K. Church and P. Hanks. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29, 1990.

[5] T. E. Dunning. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74, 1993.

[6] A. Joshi and R. Motwani. Keyword generation for search engine advertising. pages 490–496, Los Alamitos, CA, USA, 2006. IEEE Computer Society.

[7] J. Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991.

[8] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986.