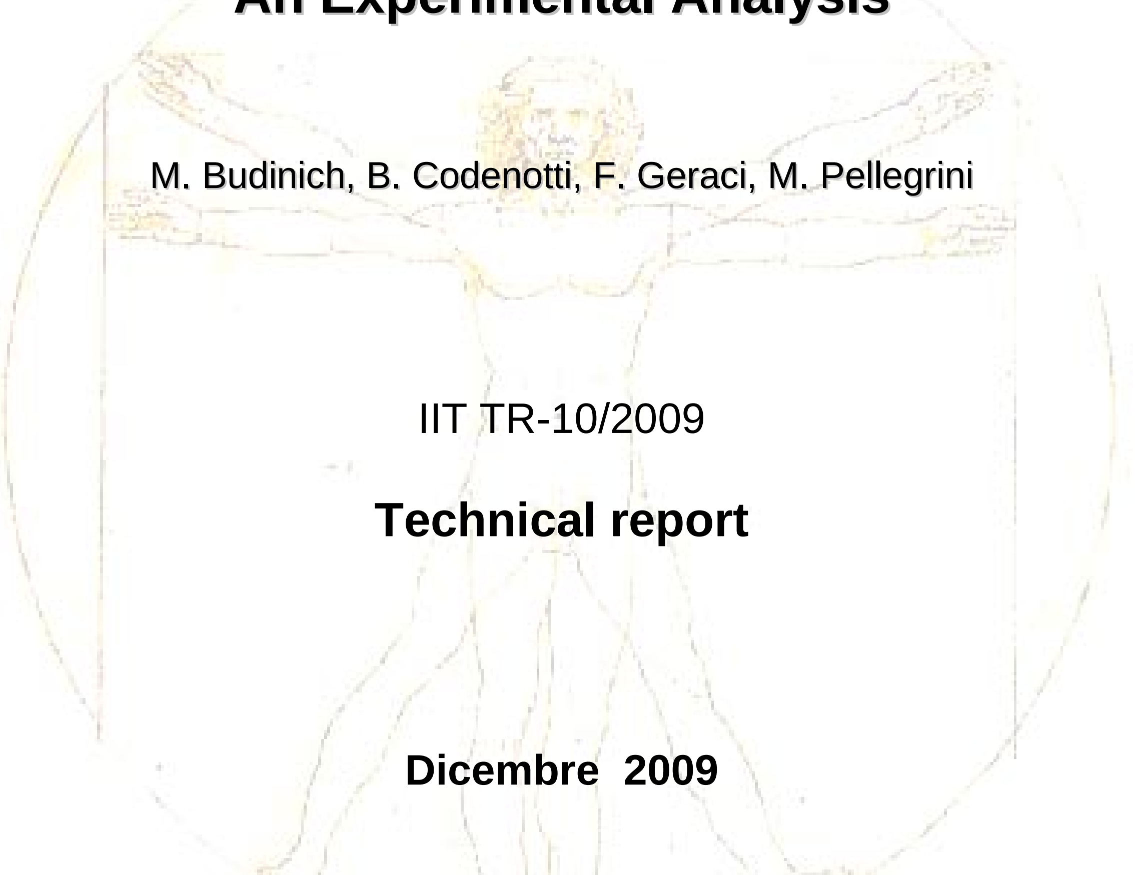
Consiglio Nazionale delle Ricerche

On the Benefits of Keyword Spreading in Sponsored Search Auctions: An Experimental Analysis



M. Budinich, B. Codenotti, F. Geraci, M. Pellegrini

IIT TR-10/2009

Technical report

Dicembre 2009

Istituto di Informatica e Telematica

On the Benefits of Keyword Spreading in Sponsored Search Auctions: An Experimental Analysis

Michele Budinich[*]
m.budinich@imtlucca.it

Bruno Codenotti[†]
bruno.codenotti@iit.cnr.it

Filippo Geraci [†]
filippo.geraci@iit.cnr.it

Marco Pellegrini [†]
marco.pellegrini@iit.cnr.it

September 30, 2009

Abstract

Sellers of goods or services wishing to participate in sponsored search auctions must define a pool of keywords that are matched on-line to the queries submitted by the users to a search engine. Sellers must also define the value of their bid to the search engine for showing their advertisements in case of a query-keyword match. In order to optimize its revenue a seller might decide to substitute a keyword with a high cost, thus likely to be the object of intense competition, with sets of related keywords that collectively have lower cost while capturing an equivalent volume of user clicks. This technique is called *keyword spreading* and has recently attracted the attention of several researchers in the area of sponsored search auctions. In this paper we describe an experimental benchmark that through large scale realistic simulations allows us to pin-point the potential benefits/drawbacks of keyword spreading for the players using this technique, for those not using it, and for the search engine itself. Experimental results reveal that keyword spreading is generally convenient (or non-damaging) to all parties involved.

1 Introduction

A very large fraction of consumers use search engines to find information on the web about goods and services before deciding whether to purchase them on the online markets. Search engines take advantage of their key position on the Web to sell advertising space to economic players on search result pages. Indeed, over the last few years, sponsored search advertising has become the dominant source of profits for search engines. Typically sponsored search results appear in two separate parts of the page above and to the right of the results returned by a search engine. Sponsored search results include a title, a short text, and a link referring to a Website. Advertising space comes in the form of slots, which are sold by auctions. When a user submits a given keyword in a query to a search engine, an auction is run among all the advertisers submitting bids for that keyword. The advertisers who wish to display their ads against the search for a keyword participate in the auction by specifying their valuation and a daily budget to the search engine. The search engine could use various mechanisms for determining winners and payments, the most popular mechanism being the generalized second price (GSP) auction.

Although GSP looks similar to the classical Vickrey-Clarke-Groves (VCG) mechanism [20, 4, 8], its properties are very different, i.e., truth-telling is not an equilibrium in GSP [6, 19]. Over the last years, several papers of computational flavor have appeared, touching in different ways this paradigm of online advertising, see, e.g., [2, 3, 6, 10, 11].

[*] Institute for Advanced Studies (IMT), Lucca, Italy.

[†] Istituto di Informatica e Telematica (IIT-CNR), Pisa, Italy.

From the viewpoint of a search engine, the *adword problem* consists of assigning a sequence of search keywords to a set of competing bidders, each with a daily spending limit, with the goal of maximizing the revenue generated by these keyword sales. This problem generalizes on-line matching, and this connection has been exploited in [13].

A central problem in adword markets from the point of view of a seller of goods and services is the generation of keywords. Advertisers typically prefer to bid for keywords that have high search volumes; however they may be very expensive, so that it might be reasonable to bid instead for several related and low volume, inexpensive terms that generate roughly the same amount of traffic altogether. Some preliminary work exploring this idea has been done in [1], where however, the emphasis is on the algorithmic aspects of keyword generation, not on the global market phenomena as in the present work.

In this paper we describe a large scale simulator for analyzing the effect of using synonyms for keyword spreading, and collect a number of evidences about the effects of this strategy. Our simulations involve up to 2M agents bidding for words from a pool of 36K words and 3M queries per experiment (more details in Sections 2 and 3). Our experiments point to the following conclusions:

- using synonyms is generally convenient to all players in the market (Figs. 6, 7a, 7b); in particular the agents using this strategy benefit the most (Figs. 7a, 7b)
- using a VCG payment scheme decreases the agents' benefits with respect to using GSP (Figs. 9, 10) while not much changes for the search engine (Fig. 8)
- as the fraction of agents using synonyms increases, the search engine revenues are not significantly affected (Fig. 11) as well as the costs for the agents not using them (Fig. 12a) while the agents using synonyms have decreasing gains (Fig. 12b)
- the budget depletion strategies are convenient for the search engine (Fig. 13)

A problem related to *keyword spreading* is that of *keyword selection* [18], where the economic players try to select at fixed rounds the subset of keywords that maximize revenues while trying to learn basic parameters (such as keyword click-through rates) during the repeated bidding processes. Note that here the viewpoint is that of a single player and that the the market, as seen by the seller, is modeled via (known or unknown) time varying probability distributions. In contrast, in our simulations keywords are selected by the agents off-line. We simulate directly the market and the auctions by using a large number of atomic agents each performing simple actions. Previous research on agent-based simulation of adwords markets by Mizuta and Steiglitz [14] was centered on studying the interaction of different classes of players according to their bidding time profiles, e.g. early vs late bidders. Kitts and LeBlanc [9] describe a large scale simulator for adwords markets to investigate several bidding strategies, e.g. random bidding vs. bid to keep relative position, which however do not involve keyword spreading. To the best of our knowledge this is the first large-scale agent-based simulation of the market effects of keyword spreading.

The rest of this paper is organized as follows. In Section 2 we briefly describe the clustering technique used to build a dictionary of synonyms. In Section 3 we highlight the architecture of our market simulator. In Section 4 we show the main outcomes of our experiments.

2 Keyword Spreading

We explore two alternative ways of performing keyword spreading. One uses the well known Wordnet ontology, the second is based on clustering web pages related to a query as found by a generalist search engine (in our case Google). The two resulting word distributions are different but the measured trends are consistent for both data sets, thus giving high confidence in the robustness of the experimental benchmark.

Wordnet The most important project for ontologies of words is *WordNet* [16]. Originally proposed by the Cognitive Science Laboratory at Princeton University only for the English language, WordNet has become a reference for the whole information retrieval community, and similar projects are now available in many other languages. WordNet is a handmade semantic lexicon that groups words into sets of synonyms called *synsets*. Intuitively one can replace a word in a text with another from the same synset without changing its semantics. A word can appear in more than one synset if it has more than one meaning. Synsets are arranged as nodes in a graph such that there is an edge to connect two nodes if there is a relation between the two synsets. There are different types of possible relations, an exhaustive list of them can be found in the WordNet web site [15]. Given two synsets X and Y, the most common types of relations in WordNet are: *hypernym* if every X is a "kind of" Y, *hyponym* if Y is a "kind of" X, *holonym* if X is a part of Y and *meronym* if Y is a part of X. In our experiments we took into account only hypernym.

Clustering Google Data Given a query word, our goal is to find a set of semantically related words whose cost is lower than those of the query. We are not only interested to paradigmatic similarity, i.e., when two words may be mutually exchanged without effects on the semantics of the text, but also to syntagmatic similarity, i.e., when two words significantly co-occur in the same context. To achieve this goal we approach the problem as a *word clustering* [5, 12] task. Given a set of objects, clustering attempts to create a partition such that the objects in a cluster are related among them, while objects in different clusters are unrelated.

Word clustering requires a corpus of documents related to the query word. To set up such a corpus we redirect the query to *Google* and download pages related to the first 100 results. Each page is later parsed and split to extract a set of sentences. Under the well established hypothesis that co-related words are more likely to stand in the same sentence, all the sentences not containing the query are discarded. We remove from each sentence over-represented words (stop words) that are often "syntactic sugar" and their removal does not affect the semantic content of the sentence. We added to the standard stop word list, a set of words that normally can not be considered stop words, but in the Web environment are considered generic (e.g. "download"). Once filtered, all the sentences are arranged in a term-document matrix whose rows correspond to sentences and whose columns to terms of the corpus. We tested different weighting schemes for terms, and we found that for our purpose a simple binary weighting scheme suffice.

For clustering we employed a fast implementation of the FPF clustering algorithm [7]. As distance between pairs of words, i.e., columns of the term-document matrix, we used the well known cosine similarity. FPF is an iterative algorithm. It makes a new cluster at each iteration and populates it by extracting from the other clusters all the elements that are more related to the new cluster. The procedure stops when a given number k of clusters is reached. In our case it is impossible to predict a good value for k. Thus, instead of feeding k in advance, we make FPF check at each iteration the number of elements in the cluster containing the query word. When this number gets below a certain threshold (10 in our case) the algorithm stops and returns the list of the words in the cluster containing the query. This procedure ensures that we find a coherent cluster of words even if the query is not central in that cluster.

3 The Simulator

The starting point in designing the simulator was the collection of some publicly available data on ad auctions:

- a reasonable set of words,
- an estimate of the cost of each word,
- an estimate of the number of clicks received by each word.

The Word List The simulator uses a finite set of words; these words represent all the possible queries that a user can make at the search engine and also all the possible keywords an advertiser can bid on. The core of the word list has been taken from the SCOWL[1] project (an open source project that maintains a set of word lists for use by spell checkers), and consists of 35867 entries.

The Traffic Estimator Google has an on-line tool (the AdWords Traffic Estimator Sandbox,[2]) developed to aid advertisers in their campaigns. The Traffic Estimator, given a keyword, displays its estimated cost per click (CPC) and the estimated number of clicks per day. The simulator uses this data to estimate some quantities that would otherwise be difficult to generate realistically. Although, as Google itself warns, the data is to be considered only as a guideline, it is of great help for our purposes.

The estimated CPC is used in the simulator (averaging the two values given by the Traffic Estimator) as a basis to assign a "real" value to each keyword. The simulator successively employs these values as parameters to generate the agents' bids and valuations. Clearly the estimated CPC of a term is different from its "real" value. If we were to measure the estimated CPCs in the simulator at the end of a run they would certainly be different from the ones supplied by the Traffic Estimator. Nonetheless their distributions and main features would be similar, and that is enough for the use we make of it.

The other values which are central to the simulator are the estimated number of clicks per day. Since the simulation considers only the queries that give rise to a click, we can simply consider the estimated number of clicks per day as the distribution of the queries in the simulator. We collected such data for each of the 35867 entries in our dictionary, building a small database that constitutes our initial data set. Table 1a summarizes the main characteristics of the data set.

Number of words	35867
Max. clicks per day	349216
Min. clicks per day	1
Max. CPC	$23.6
Min. CPC	$0.05

(a) figures from the data set

	clustering	Wordnet
Words with synonyms	18660	12271
Max. synonyms for a word	13	441
Max. terms a word is synonym of	668	146

(b) figures from the synonyms databases

Table 1: Main data from the word and synonym databases.

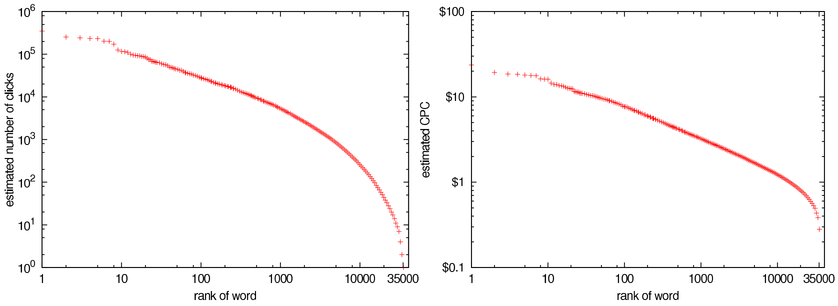
For completeness, we plotted the data collected from Google's Traffic Estimator. Figure 1a is the distribution of the estimated clicks per day, while Figure 1b shows how estimated average costs per click are distributed.

We want to use our simulator to investigate the behavior of ad auction mechanism in the presence of agents who make use of keyword spreading. To model such behaviors we need a set of synonyms for each word. The clustering algorithms described in Section 2 produced a list of synonyms for each word. As a reference we have also created a similar list by querying the Wordnet[3] database. Table 1b gives some basic figures on the two resulting data sets, while Figure 2a and Figure 2b show the distribution of the number of synonyms per word and the distribution of the number of terms a word is synonym of. There is a big difference in the boundary values of the databases: for example, there is a term for which Wordnet gives 441 synonyms; but more important is the difference in the rank distribution (see Fig. 2a). Due to limitations in the computational resources, the clustering imposed a hard limit of 13 on the maximum number of synonyms per word. Nonetheless, as shown

[1] `http://wordlist.sourceforge.net/`

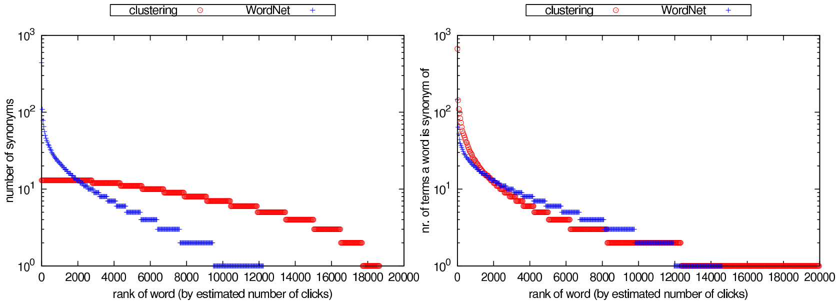
[2] `https://adwords.google.com/select/TrafficEstimatorSandbox`

[3] `http://wordnet.princeton.edu/`



(a) estimated number of clicks **(b)** estimated CPC

Figure 1: Data gathered from Google's Traffic Estimator, in log-log scale.



(a) number of synonyms per word **(b)** number of words a term is a synonym of

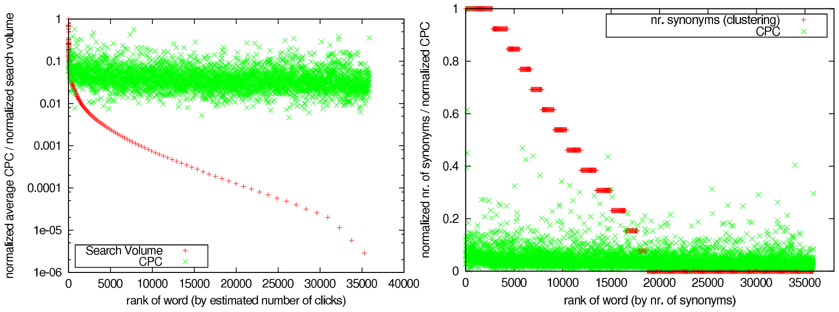
Figure 2: Comparison of the two synonyms databases in log-scale.

clearly by Figure 2a, the majority of the words have more synonyms in the clustering database than in the Wordnet one. Overall we can consider the databases comparable for our purposes, and the experimentally detected trends are consistent in both databases.

Starting from a list of words, we have expanded it with various information: prices, number of clicks and synonyms. It seems now a natural question to ask if there is any correlation between these quantities. As a first guess it might seem reasonable to expect at least some correlation. That is, we might expect that some "popular" words receive many clicks and have a high price. Or that words that receive a lot of clicks also happen to have many synonyms. Somewhat surprisingly, an empirical analysis gives a negative result. At a first glance the data set exhibits virtually no correlation between the different values. To give a rough idea of this result we present just two plots, all the other ones being extremely similar. Figure 3a ranks words by estimated number of clicks, and shows these values along the CPCs (both normalized). It looks like there is no order in the CPC values; they appear as if uniformly distributed. Figure 3b, instead, ranks the words by the number of synonyms they possess,[4] and displays this value along the estimated CPC (again, normalized). As it seems apparent there is no correlation between these quantities. All the other comparisons, e.g. CPCs versus click volume, synonyms versus CPC using the clustering database, give similar results.

We are now ready to describe our simulator. The simulator has a very simple structure. It keeps the agents and the words stored in two arrays. Each word contains a pointer to a balanced binary

[4] Using the Wordnet database.



(a) nr. of clicks and CPC (log-scale)

(b) nr. of synonyms and CPC

Figure 3: Comparison of the two synonyms databases used.

search tree[5] that contains the bids on that keyword. Along the bids, each node of the tree stores a pointer to the agent that placed that bid.

All the simulations were carried out using the same static set of agents. To this end, the set of agents was generated once and for all and saved in a file. Its main characteristics are presented in Table 2a. In what follows we will refer to this fixed set of agents, words and synonyms as our data set.

Number of Agents	$2 \cdot 10^6$
Max. bids on a single word	21446
Min. bids on a single word	21
Max. bids per agent	3000
Min. bids per agent	3
Budget Range	[\$1 – \$100]
Bid Range	[\$0.01 – \$200]
Nr. of slots	4
Clickthrough probabilities	$0.6, 0.25, 0.10, 0.05$

(a) figures on generated agents

Keyword	"reviews"
"Real" value	\$1.045
Estimated nr. of clicks	12029
Nr. of interested agents	11023
Max. bid	\$3.064
Min. bid	\$0.010
Max. difference $v_i - b_i$	12.5% of v_i

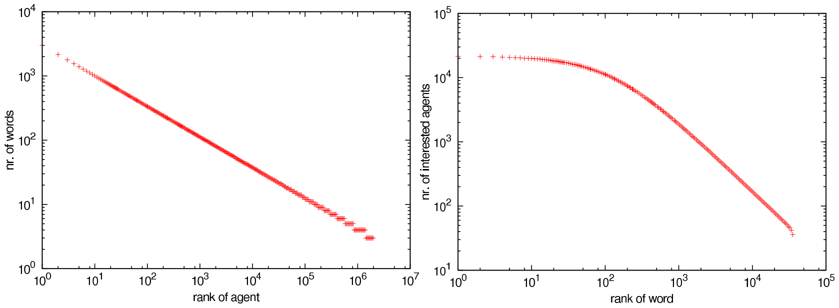
(b) details about a sample keyword

Table 2: Figures from standard dataset.

Each agent bids on a number of different keywords. If we consider all the agents, these values are distributed as a power law, whose parameters are based on the number of agents, such as to keep a fixed maximum and minimum (to avoid cases in which an agent bids on all of the words, or cases in which there are agents that have not bid on any word at all). Figure 4a plots these values for the data set.

Another quantity characterizing agents is their budget. In this case a uniform distribution has been chosen, and the budgets are all in the [\$1 – \$100] range.

[5] Specifically an AVL tree.

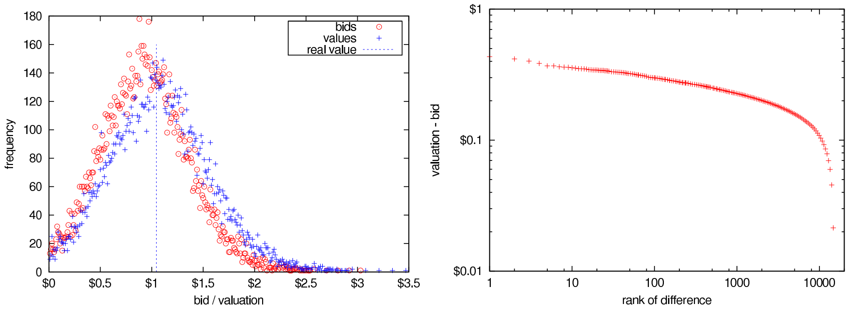


(a) words per agent **(b)** agents per word

Figure 4: Agents per word and words per agent in log-log scale.

Words Having fixed the number of words an agent will bid on, the next step is to select them from the dictionary. The simulator does so, and the resulting values (i.e., the number of agents interested in every word) is again distributed as a (different) power law. The parameters controlling such distribution are chosen as to avoid unrealistic scenarios. Figure 4b shows the number of interested agents per word in our data set.

As described above, each word is assigned a "real" value based on the data gathered from the Traffic Estimator. Based on this reference value, each agent i will then compute its personal valuation v_i for the keyword. The distribution of the valuations for each agent is a normal distribution whose mean is precisely the "real" value of the word. To increase the variety among agents, each agent has a different variance associated to this normal distribution. Figure 5a shows the the distribution of valuations for different agents interested in the same keyword, i.e. "reviews".



(a) bids and valuations **(b)** difference valuations - bids

Figure 5: The bids and valuations for a single word, i.e. "review", whose actual value is 1.045.

As a final step each agent i must generate a bid b_i. Bids are generated according to the agent's valuation v_i. Only bids such that $b_i < v_i$ are considered, and they are generated so that the differences $v_i - b_i$ are distributed according to a power law. Figure 5b shows the resulting plot for the keyword "reviews". Except for the maximum number of interested agents per word, all the other characteristics described are parameters to the simulator. The number of interested agent per word is indirectly controlled by setting the maximum and minimum number of words a single agent can bid on.

4 Experimental Results

Fig. 6 shows the increment in search engine revenue between a basic simulation (in which no agent ever changes keywords) and one where we allow 20% of the agents to apply keyword spreading, using both the Wordnet database and our clustering techniques. The difference levels-off with a gain of 0.5 to 0.8 of a point. Given the total value of the adwords market this difference is very significant in absolute terms.

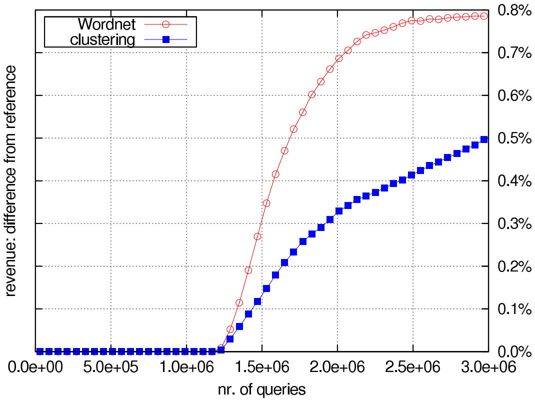
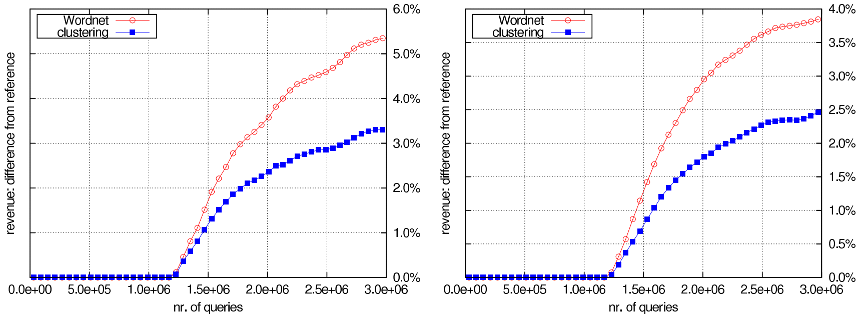


Figure 6: Comparison of the search engine's revenue when we allow some agents to change their keywords with synonyms provided by Wordnet and by the clustering technique.

In Fig. 7a we show the increase in revenue for the agents that are allowed to change words (20% of total), and in Fig. 7b for those that are not allowed (80% of total). For both groups the increase in revenue is positive and levels-off, with the agents in the fist group having better performance (in the range 3.2%-5.3%).



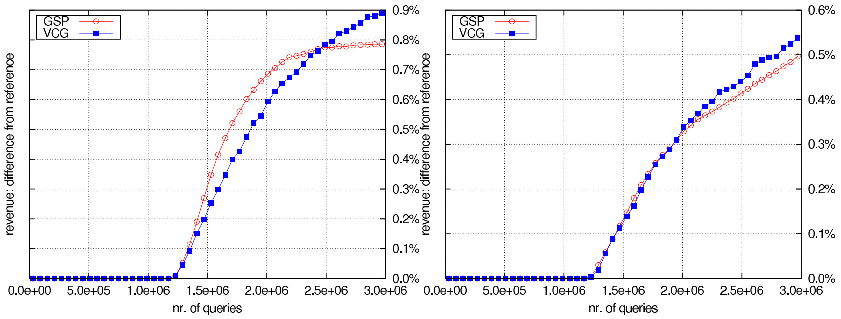
(a) keyword spreading agents

(b) non keyword spreading agents

Figure 7: Increase in revenue for agents.

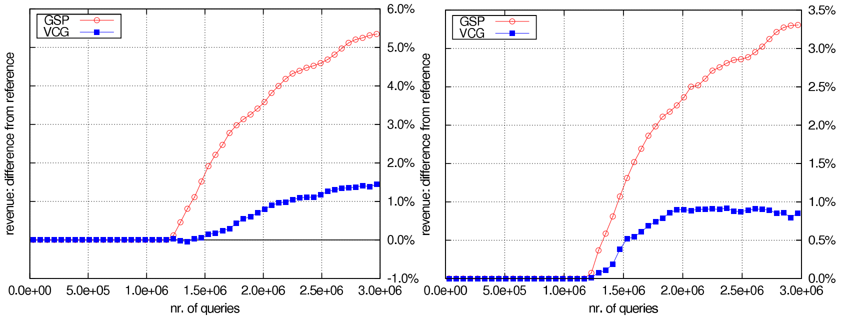
The simulations in figures 6,7a and 7b were all run under the GSP mechanism.

Figures 8, 9 and 10 report the results of experiments to test the impact of VCG and GSP policies in the revenue increment of the search engine (8), of the agents performing keyword spreading (9) and of those not using this policy (10). We note that the impact of the two policies is quite relevant for the single agents, with GSP yielding an increment more than double than that of VCG. In contrast for search engines the difference in increment is about 0.1%.



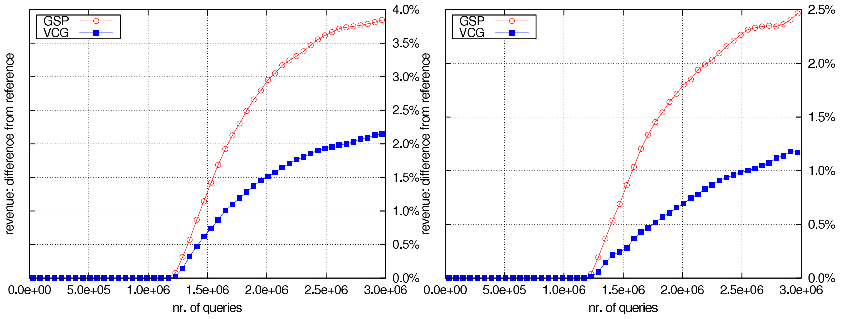
(a) Wordnet **(b)** clustering

Figure 8: Increase in revenue for the search engine under VCG and GSP pricing policies



(a) Wordnet **(b)** clustering

Figure 9: Increase in revenue for agents using keyword spreading under VCG and GSP pricing policies



(a) Wordnet **(b)** clustering

Figure 10: Increase in revenue for agents not using keyword spreading under VCG and GSP pricing policies

Figures 11, 12a, 12b show the variation in revenue increase when we vary the fraction of users using keyword spreading from 5% to 95% of the total. While the increased revenue for the search engine and for non-keyword spreading agents is hardly affected, we notice a clear effect of diminishing marginal gain for keyword-sprading agents, with initial gains up to 8% for early adopters, and just 2% when the practice in widespread.

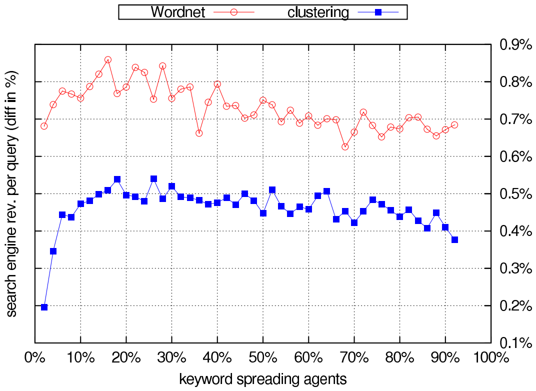
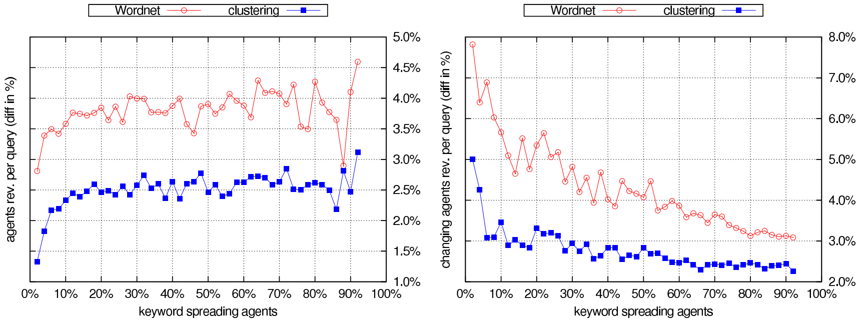


Figure 11: Search engine's increase in revenue for varying fraction of keyword spreading agents.



(a) non keyword-spreading agents

(b) keyword-spreading agents

Figure 12: Agent's increase in revenue for varying fraction of keyword spreading agents.

We also explore the effect of a class of strategic behavior called *budget depletion strategies*. Here for 20% of the words the last bidder of each word that obtains an advertisement slot will switch to a policy of increasing its bid so to deplete the competition's budget faster, without incurring in any additional cost. We explore two cases (a) called "unrealistic" where the strategic agent knows the optimal new bid value, and (b) called "realistic" where the optimal new bid value is sought by small successive increments. Figure 13 shows the percentage change in the search engine's revenue, when budget depletion strategies are used.

References

[1] Vibhanshu Abhishek, Kartik Hosanagar. Keyword Generation for Search Engine Advertising using Semantic Similarity between Terms, Proceedings of the 9th International Conference on Electronic Commerce, ICEC 2007.

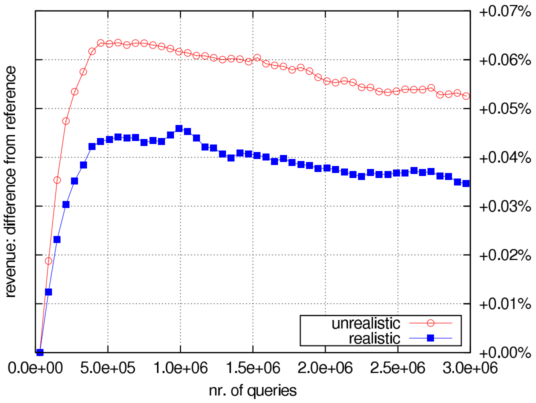


Figure 13: Increase in search engine's revenue with budget depletion strategies.

[2] Gagan Aggarwal, Ashish Goel, and Rajeev Motwani. Truthful auctions for pricing search keywords. In Proceedings of the 7th ACM Conference on Electronic Commerce, pages 1-7, 2006.

[3] Sihem Amer-Yahia, Sebastien Lahaie, and David M. Pennock. Towards a generic bidding standard for online advertising. In Fourth Workshop on Ad Auctions, 2008.

[4] E. H. Clarke. Multipart pricing of public goods. Public Choice, 11:17-33, 1971.

[5] Inderjit S. Dhillon, Subramanyam Mallela, Rahul Kumar, Enhanced word clustering for hierarchical text classification, Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 191-200 (2002.

[6] Benjamin Edelman, Michael Ostrovsky, and Michael Schwarz. Internet advertising and the Generalized Second Price auction: Selling billions of dollars worth of keywords. American Economic Review, 97(1):242-259, 2007.

[7] Teofilo F. Gonzalez, Clustering to Minimize the Maximum Intercluster Distance, Theoretical Computer Science 38 (2/3), pp.293-306 (1985).

[8] Theodore Groves. Efficient collective choice when compensation is possible. Review of Economic Studies, 46:227-241, 1979.

[9] Brendan Kitts, Benjamin J. LeBlanc: A Trading Agent and Simulator for Keyword Auctions. Proceedigns of AAMAS 2004: 228-235

[10] Sebastien Lahaie. An analysis of alternative slot auction designs for sponsored search. In Proceedings of the 7th ACM conference on Electronic commerce, pages 218-227, 2006.

[11] Sebastien Lahaie, David M. Pennock, Amin Saberi, and Rakesh V. Vohra. Sponsored search auctions. In Algorithmic Game Theory, chapter 28, pages 699-716. Cambridge University Press, 2007.

[12] Hang Li and Naoki Abe, Word clustering and disambiguation based on co-occurrence data, Proceedings of the 17th international conference on Computational linguistics, pp. 749–755 (1998).

[13] A. Mehta, A. Saberi and U. Vazirani. Adwords and Generalized Online Matching, Journal of ACM 54(5), 2007.

[14] Mizuta, H. and Steiglitz, K. 2000. Agent-based simulation of dynamic online auctions. In Proceedings of the 32nd Conference on Winter Simulation (Orlando, Florida, December 10 - 13, 2000). Winter Simulation Conference. Society for Computer Simulation International, San Diego, CA, 1772-1777.

[15] George A. Miller, Christiane Fellbaum, Randee Tengi, Pamela Wakefield, Rajesh Poddar, Helen Langone, Benjamin Haskell, WordNet: A Lexical Database for English, Princeton University (2006).

[16] George A. Miller, WordNet: An On-line Lexical Database, International Journal of Lexicography (1990)

[17] Matthew Richardson, Ewa Dominowska, and Robert Ragno. Predicting clicks: estimating the click-through rate for new ads. In Proceedings of the 16th Inter- national World Wide Web Conference, pages 521-530, 2007.

[18] Paat Rusmevichientong, David P. Williamson: An adaptive algorithm for selecting profitable keywords for search-based advertising services. ACM Conference on Electronic Commerce 2006: 260-269

[19] Hal R. Varian. Position auctions. International Journal of Industrial Organiza- tion, 25:1163-1178, 2007.

[20] William Vickrey. Counterspeculation, auctions and competitive sealed tenders. Journal of Finance, 16:8-37, 1961.