ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

PH.D. THESIS

Automated Creation and Optimization of Online Advertising Campaigns

Stamatina Thomaidou

Department of Informatics

Athens, October 2014

Preface

This PhD thesis is submitted to the Athens University of Economics and Business in partial fulfilment of the requirements for the degree of Doctor of Philosophy. The PhD project has been performed at the Department of Informatics of Athens University of Economics and Business. The thesis committee members are:

- Michalis Vazirgiannis (Advisor), Professor at Athens University of Economics and Business
- Ion Androutsopoulos, Associate Professor at Athens University of Economics and Business
- Kjetil Nørvåg, Professor at Norwegian University of Science and Technology (NTNU)

This research has been co-financed by the European Union (European Social Fund - ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF)- Research Funding Program: Heracleitus II. Investing in knowledge society through the European Social Fund.

European Union
European Social Fund
OPERATIONAL PROGRAMME
EDUCATION AND LIFELONG LEARNING
investing in knowledge society
MINISTRY OF EDUCATION & RELIGIOUS AFFAIRS, CULTURE & SPORTS
MANAGING AUTHORITY
Co-financed by Greece and the European Union
NSRF
2007-2013
programme for development
EUROPEAN SOCIAL FUND

Automated Creation and Optimization of Online Advertising Campaigns

Abstract

In this thesis, we tackle emerging issues of online advertising (sponsored search, textual ads, online advertising campaigns). The inherent competitiveness of the paid search market and the fact that an advertising campaign development is a laborious task involving significant human resources and expertise have led to the need of designing a system which will automatically create and optimize online advertising campaigns. We introduce the discussion of these issues from the perspective of the advertiser role and not of the auctioneer's, as one of our main novelties. In this context, our purpose is twofold. We aim to propose a methodology as well as a functional prototype system for automated creation, monitoring, and optimization of cost-efficient pay-per-click campaigns with budget constraints.

The research areas of sponsored search, textual advertising and keyword research address challenges in automatic extraction, suggestion, and expansion of keywords as well as finding an optimal bidding strategy from the advertiser's perspective. In addition, an open problem is the automated ad creative generation process. Motivated by the existing literature and directions, we propose a novel framework that, given a landing page in the context of the promotion of products and services, automates the complete life cycle of a campaign. Our framework automatically extracts and suggests bidding keywords for online advertising campaigns as well as automatically generates advertisement texts. Furthermore, it manages and optimizes the bidding settings of the campaign.

Keyword selection is one of the most important success factors for online advertising. In the online advertising campaign platforms, the bidding keywords are actually keyphrases, thus higher order n-grams and not only unigrams. The major problem of an advertising campaign that takes into account only the suggestions of the most popular queries is that they are widely used, which means the proposed keywords are quite competitive and expensive. The other problem is that they are volume-based (i.e., very generic terms), which means these

keywords will tend to drive more traffic without guarantee any user actions on the landing page. Thus, to face this problem the system extracts terms from a given landing page and then generates additional keywords that are highly relevant and specific yet non-obvious to some of the existing terms inside the webpage with less competition (i.e., lower bidding values).

Considering the problem of ad-text generation, we introduce a novel method that produces in an automatic manner compact text ads (promotional text snippets), given as input a product description webpage (landing page). The challenge in this problem is to produce a small comprehensive ad while maintaining at the same time relevance, clarity, and attractiveness. Our method follows a pipeline approach. Initially, it formulates relevant and important n-grams given the landing page. We continue with transforming them into snippets and we have built a language model trained on ads to evaluate phrases in terms of their marketing appeal. In addition, the snippets must have a positive meaning in order to have a call-to-action style, thus we use sentiment analysis on them.

We articulate the budget optimization problem as a multiple-choice knapsack for which we find the most profitable combination of keywords and their bids. We approximate the solution capitalizing on a genetic algorithm for budget optimization with multiple keyword options. We also propose the use of keyword statistics to predict keyword behavior using multiple linear regression. In this way, the optimization module focuses on the learning process from existing campaign statistics and also from applied strategies of previous periods in order to invest optimally in the next period. The objective is to maximize the performance (i.e., clicks or actions) under the current budget constraint.

Our proposed framework (methodologies and prototype system) is experimentally evaluated not only on simulated environment but also on real world campaigns. Through different scenarios, we demonstrate that our framework presents a promising behavior with regards to campaign performance statistics as it outperforms systematically the competing manually maintained campaigns, assisting effectively in this way the advertiser.

Αυτοματοποιημένη Δημιουργία και Βελτιστοποίηση
σε Διαδικτυακές Διαφημιστικές Καμπάνιες

Περίληψη

Αντικείμενο της παρούσας διατριβής είναι η αντιμετώπιση αναδυόμενων ζητημάτων της διαδικτυακής διαφήμισης (χορηγούμενη αναζήτηση, διαφημιστικά κείμενα, διαδικτυακές διαφημιστικές καμπάνιες). Η έμφυτη ανταγωνιστικότητα της αγοράς των διαδικτυακών διαφημίσεων και προώθησης μέσω των μηχανών αναζήτησης, καθώς και το γεγονός πως η ανάπτυξη μίας διαδικτυακής διαφημιστικής καμπάνιας είναι μία πολύπλοκη διεργασία που συνεπάγεται σημαντικό ανθρώπινο δυναμικό και τεχνογνωσία, οδήγησαν στην ανάγκη σχεδιασμού ενός συστήματος όπου θα δημιουργεί και θα βελτιστοποιεί αυτόματα διαδικτυακές διαφημιστικές καμπάνιες. Εισάγουμε τη συζήτηση αυτών των ζητημάτων από τη σκοπιά του ρόλου του διαφημιστή και όχι του διοργανωτή των δημοπρασιών διαφήμισης, ως μία από τις κύριες καινοτομίες μας. Στο πλαίσιο αυτό, ο σκοπός μας είναι διττός. Στόχος μας είναι να προτείνουμε μία μεθοδολογία καθώς και ένα λειτουργικό πρότυπο σύστημα για την αυτόματη δημιουργία, παρακολούθηση και βελτιστοποίηση σε καμπάνιες με περιορισμούς στον διαθέσιμο προϋπολογισμό.

Οι ερευνητικές περιοχές της χορηγούμενης αναζήτησης, των διαφημιστικών κειμένων και της επιλογής διαφημιστικών όρων (λέξεις-κλειδιά) αντιμετωπίζουν προκλήσεις στην αυτόματη εξαγωγή, πρόταση και επέκταση των λέξεων-κλειδιών, καθώς και την εξεύρεση της βέλτιστης στρατηγικής πλειοδοτήσεων από την πλευρά του διαφημιστή. Επιπλέον, ένα ανοικτό πρόβλημα είναι η αυτοματοποιημένη διαδικασία δημιουργίας μικρών διαφημιστικών κειμένων. Ορμώμενοι από την υπάρχουσα βιβλιογραφία και κατευθύνσεις, προτείνουμε μία καινοτομική δομή και σύστημα όπου δεδομένης μια σελίδα προορισμού, στο πλαίσιο της προώθησης προϊόντων και υπηρεσιών, αυτοματοποιεί τον πλήρη κύκλο ζωής μίας καμπάνιας. Το σύστημά μας εξάγει αυτόματα και προτείνει λέξεις-κλειδιά για διαδικτυακές διαφημιστικές καμπάνιες, όπως επίσης και παράγει αυτόματα μικρά διαφημιστικά κείμενα. Επιπροσθέτως, διαχειρίζεται και βελτιστοποιεί τις ρυθμίσεις πλειοδότησης της καμπάνιας.

Η επιλογή των λέξεων-κλειδιών είναι ένας από τους πιο σημαντικούς παράγοντες επιτυχίας για τη διαδικτυακή διαφήμιση. Στις διαδικτυακές διαφημιστικές πλατφόρμες, οι λέξεις-κλειδιά είναι στην πραγματικότητα φράσεις, που σημαίνει ότι μπορεί

να αποτελούνται όχι μόνο από μία αλλά και παραπάνω λέξεις. Το κύριο πρόβλημα μίας διαφημιστικής καμπάνιας που λαμβάνει υπόψη μόνο τις προτάσεις από τις πιο δημοφιλείς επερωτήσεις είναι ότι χρησιμοποιούνται ευρέως και από τους ανταγωνιστές-διαφημιζόμενους. Αυτό σημαίνει την επιστροφή και πρόταση πολύ ανταγωνιστικών και συνεπώς ακριβών λέξεων-κλειδιών. Το άλλο πρόβλημα που προκύπτει όταν βασίζονται στη συχνότητα αναζήτησής επερωτήσεων από χρήστες (δηλ. πολύ γενικοί όροι), είναι η τάση αυτών των λέξεων-κλειδιών να δημιουργήσουν μεγαλύτερη επισκεψιμότητα, χωρίς όμως ταυτόχρονα να εγγυώνται κάποια ενέργεια μέσα στη σελίδα από τους χρήστες. Για να αντιμετωπίσει αυτό το πρόβλημα, το σύστημα εξάγει όρους από μία δοθείσα ιστοσελίδα του διαφημιζόμενου και στη συνέχεια δημιουργεί και προτείνει περαιτέρω λέξεις–κλειδιά πολύ σχετικές με τη συγκεκριμένη σελίδα αλλά ταυτόχρονα όχι γενικές και πολύ εμφανείς ώστε να κοστίζουν λιγότερο.

Λαμβάνοντας υπόψη το πρόβλημα της παραγωγής διαφημιστικών κειμένων, εισάγουμε μία καινοτομική μέθοδο που παράγει με αυτόματο τρόπο διαφημίσεις με πολύ μικρό και συμπαγές κείμενο, δοθείσας μίας ιστοσελίδας με περιγραφή του διαφημιζόμενου προϊόντος (σελίδα προορισμού). Η πρόκληση σε αυτό το πρόβλημα είναι να παραχθεί ένα μικρό κείμενο με συμπυκνωμένη πληροφορία διατηρώντας παράλληλα σχετικότητα, σαφήνεια, και ελκυστικότητα. Η μέθοδός μας ακολουθεί μία προσέγγιση σωλήνωσης. Αρχικά, κατασκευάζει σχετικά και σημαντικά ν-γράμματα (*ngrams*). Συνεχίζουμε με την μετατροπή τους σε μικρές προτάσεις και έχουμε δημιουργήσει ένα μοντέλο γλώσσας εκπαιδευμένο σε διαφημίσεις για την αξιολόγησή τους από την άποψη της ελκυστικότητας. Επιπλέον, οι διαφημίσεις πρέπει να έχουν ένα θετικό μήνυμα, προκειμένου να έχουν μία πειστική μορφή, οπότε χρησιμοποιούμε ανάλυση συναισθήματος.

Διατυπώνουμε το ζήτημα της βελτιστοποίησης του προϋπολογισμού ως πρόβλημα πολλαπλής επιλογής σακιδίου για την οποία βρίσκουμε τον πιο κερδοφόρο συνδυασμό λέξεων-κλειδιών και των πλειοδοτήσεών τους. Προσεγγίζουμε τη λύση βασιζόμενοι σε ένα γενετικό αλγόριθμο για τη βελτιστοποίηση του προϋπολογισμού με πολλαπλές επιλογές λέξεων-κλειδιών . Προτείνουμε επίσης τη χρήση στατιστικών των λέξεων-κλειδιών για να προβλέψουμε την απόδοσή τους με τη χρήση πολλαπλής γραμμικής παλινδρόμησης. Με αυτόν τον τρόπο, το κομμάτι της βελτιστοποίησης εστιάζει στη διαδικασία μάθησης από τα υπάρχοντα στατιστικά στοιχεία της καμπάνιας, αλλά και από εφαρμοσμένες στρατηγικές προηγούμενων περιόδων προκειμένου να επενδύσουμε βέλτιστα στην επόμενη περίοδο. Ο στόχος είναι η μεγιστοποίηση της απόδοσης (π.χ. κλικάρισμα ή ενέργεια μέσα στη σελίδα) υπό τον περιορισμό του προϋπολογισμού.

Το προτεινόμενο πλαίσιό μας (οι μεθοδολογίες και το πρότυπο σύστημα) αξιολογείται πειραματικά όχι μόνο σε προσομοιωμένο περιβάλλον, αλλά και σε πραγματικές καμπάνιες. Μέσα από διαφορετικά σενάρια, αποδεικνύουμε ότι το πλαίσιό μας παρουσιάζει μία αποδοτική συμπεριφορά σε σχέση με τα στατιστικά στοιχεία απόδοσης της καμπάνιας, καθώς ξεπερνά συστηματικά τις ανταγωνιστικές διαχειριζόμενες χειρονακτικά καμπάνιες, βοηθώντας αποτελεσματικά με αυτόν τον τρόπο τον διαφημιστή.

Acknowledgements

First and foremost I would like to express my appreciation and gratitude to my supervisor Michalis Vazirgiannis for trusting me and supporting me throughout the years of my PhD. He encouraged me from our very first collaboration in my MSc studies to pursue the PhD in the inspiring field of Data Mining. I would also like to thank the members of my doctoral committee, Ion Androutsopoulos, Kjetil Norvag, Iraklis Varlamis, George Giannakopoulos, Maria Halkidi, and Yannis Theodoridis for their thorough reviews and comments of this PhD thesis.

Special thanks go to my fellow colleagues in AUEB, Margarita, Vaggelis D., Xenofon, Nikos F., Christos, Nikos S., and Elli. All these years, they have been an amazing support and influence. Especially, Margarita, apart from a close friend, was always there for helping me and motivating me at every step of this journey. Vaggelis guided me many times with inspiring conversations. Many thanks also to Konstantinos, Ismini, Vaggelis A., and Panagiotis for their enthusiastic contribution in my PhD research. Moreover, I would like to thank all the members from AUEB DB-Net and NLP research groups as well as the DaSciM group of École Polytechnique.

I am grateful to all of my friends that were beside me during my MSc and PhD research, especially Eleni, Maria, Christina, Manolis, Kyriakos, and Dimitris. No words in any language would be sufficient to thank Makis for being part of my life, and for supporting me to all my decisions and my PhD journey.

Last, but definitely not least, I would like to thank my parents, Konstantina and Kyriakos, and my brother, Manolis, for their continuous support and love. They have been always and they will be always my true inspiration.

Στον Μάκη

Contents

Chapter 1

Introduction

1.1 Motivation

The advertising industry is rapidly changing as companies, individuals, and advertisers increasingly understand the usefulness and value of the World Wide Web as an integral part of our lives and as a more targeted mean of promotion. The audience is already looking for what the advertiser wants to sell. This is translated to better return of investment (ROI) for the company, since the money spent is directly reaching potential buyers and not uninterested audience. It is significantly cheaper than traditional advertising because of this targeted nature. The prevalent pay-per-click model lets you pay only when someone chooses to see you. Finding the correct search keywords to advertise on and follow a strategy that learns the proper prices of web-based advertising space which are dependent on the relevance of the surrounding web content and the traffic that the website receives can lead to profit generation. In general, online advertising is gaining acceptance and market share while it has evolved into a $26 billion industry for advertisers[1]. There are many different types of online advertising including contextual ads on search engine result pages, banner ads, rich media ads, social network advertising, advertising networks and e-mail marketing. In this thesis, we will discuss about the *sponsored search advertising case*.

[1] http://www.iab.net/AdRevenueReport

One form of online advertising is the promotion of products and services through search-based advertising. This follows an ad auction process with pay-per-click (PPC) model [26] for the advertisers. The selected search engine has the role of the auctioneer for the ad slots and the bids that the advertisers set for keywords and their ads. The dominant strategy for ad selection is the hybrid second-price auction [27] system. The three most prevalent options in the search-based advertising market are Google AdWords, Yahoo Search Marketing, and Microsoft AdCenter (the two latter have merged) [2]. Today's most popular search-based advertising platform is Google AdWords having the largest share of revenues amongst its competitors. Google [3] 2010 annual report showed that company's advertising revenues made up 97% of its revenues in 2008 and 2009, and 96% of its revenues in 2010. In 2010, Google advertising revenues (as an auctioneer) were $ 29 billion [4]. Search remains the largest online advertising revenue format, accounting for 46.5% of 2011 revenues, up from 44.8% in 2010. In 2011, Search revenues totaled $14.8 billion, up almost 27% from $11.7 billion in 2010. Search has remained the leading format since 2006, having strong sequential growth[5].

Effective keyword selection is one of the most important success factors for online advertising. Companies would like to advertise on the most effective keywords to attract only prospective customers and not uninterested browsing users. In addition, they need well-written ad creatives to attract more visitors and generate thus higher revenues. The major problem of an advertising campaign that takes into account only the suggestions of the most popular terms is that they are widely used, therefore the relevant keywords are quite competitive in terms of cost-per-click (CPC) cost. Another issue is that they are volume-based (i.e., number of monthly searches), which means these keywords will tend to drive more traffic to the campaign but not necessarily proportional conversions.

In addition, the preparation of large scale online advertising campaigns for products, services, brands, or web pages can be a very complex task if it is designed for websites with online catalogs or catalog aggregators. The shops or

[2] `http://www.searchalliance.com/publishers`

[3] Google Search is the most popular search engine with more than 300 million searches a daily basis. This ranks Google Search as the top search engine and also the top site in terms of web-traffic - `http://www.alexa.com/topsites`

[4] `http://investor.google.com/earnings.html`

[5] `http://www.iab.net/AdRevenueReport`

listings are classified according to the products that they are selling, so each landing page contains important information and a relevant description for each category or product that needs to be considered. The number of the various URLs inside these domains makes the effort even more complicated regarding the manual insertion of keywords and ad-texts per landing page.

Finally, the most challenging problem in terms of managing an advertising campaign, is the budget optimization. In particular every advertiser needs a good strategy for selecting the most effective keywords and the amount of bidding for each, in the view of either maximizing the profits for the advertiser who is investing a limited budget or maximize the traffic on their website.

1.2 Contributions

The principal goal of this thesis is to offer an integrated approach and a fully functioning prototype system for automated advertising campaign creation, management, and monitoring for profit optimization under budget constraints. Thus, we present methods for selecting the most effective keywords, generating in an automated way ad-texts, and choosing the proper bids, aiming at maximizing either the profits for the advertiser based on a specific budget or the traffic on their website. In this effort, we focus only on the advertisers and not on the other bidders or the self-interested auctioneer. We select Google AdWords as the advertising platform and channel platform due to its dominant role in the share of web-search advertising volume.

Our main contributions are the following:

- We propose a method for recommending to the advertiser multiword terms (bigrams, trigrams) with high specificity without the need to capitalize on usage data such as query and web traffic logs.
- We propose a technique which produces compact ad-text snippets in an automated and massive manner given a product landing page as input.
- We propose an approximate solution to the budget optimization problem to maximize profit or traffic, the two usual objectives for websites.

- We present a fully implemented and functional prototype system, developed for the Google AdWords platform.
- We conduct a comprehensive experimental evaluation not only on a simulated environment but on real world campaigns as well.

Overall, the proposed framework can contribute to considerably optimizing the resources (time and experienced personnel) devoted to developing and monitoring a campaign. On top of this the monitoring module ensures the maximization of the profit respecting the available budget.

1.3 Publications

Our contributions to this PhD work have been published in several international conferences. Below is the list of our publications and the corresponding chapters where publications are based upon.

- Stamatina Thomaidou, Michalis Vazirgiannis, Kyriakos Liakopoulos. Toward an Integrated Framework for Automated Development and Optimization of Online Advertising Campaigns. Intelligent Data Analysis Journal. To Appear in Volume 18(6) [99]
  This publication is based on Chapter 6
- Stamatina Thomaidou, Ismini Lourentzou, Panagiotis Katsivelis-Perakis, Michalis Vazirgiannis. Automated Snippet Generation for Online Advertising. ACM International Conference on Information and Knowledge Management (CIKM'13), San Francisco, USA [98]
  This publication is based on Chapter 4
- Stamatina Thomaidou, Konstantinos Leymonis, Michalis Vazirgiannis. GrammAds: Keyword and Ad Creative Generator for Online Advertising Campaigns. Digital Enterprise Design & Management Conference (DED&M'13), France, Paris [97]
  This publication is based on Chapter 3

- Stamatina Thomaidou, Konstantinos Leymonis, Kyriakos Liakopoulos, Michalis Vazirgiannis. AD-MAD: Integrated System for Automated Development and Optimization of Online Advertising Campaigns. IEEE International Conference on Data Mining Workshop (ICDMW'12), Brussels, Belgium [96]
  This publication is based on Chapter 6

- Kyriakos Liakopoulos, Stamatina Thomaidou, Michalis Vazirgiannis. The Adomaton Prototype: Automated Online Advertising Campaign Monitoring and Optimization. Ad Auctions Workshop, ACM Conference on Electronic Commerce (AAW'12-EC'12), Valencia, Spain [58]
  This publication is based on Chapter 5

- Stamatina Thomaidou, Michalis Vazirgiannis. Multiword Keyword Recommendation System for Online Advertising. IEEE/ACM International Conference on Advances in Social Network Analysis and Mining (ASONAM'11), Kaohsiung, Taiwan [95]
  This publication is based on Chapter 3

1.4 Thesis Organization

The thesis is organised as follows. In Chapter 2 we offer an overview of important concepts regarding challenges and key issues in Online Advertising, including also background and state-of-the-art models. In Chapter 3 we describe the *Keyword Generation* task which is the first component of our proposed framework. In Chapter 4 we present the second component, the *Ad-Text Generation* task. In Chapter 5 we discuss and analyze the proposed strategy for Campaign Management and Optimization, presenting an approximate solution for the *Budget Optimization* problem. In Chapter 6 we give an overall presentation of the integrated *Adomaton Prototype* system as well as detailed software design specifications and use cases examples. Finally, in Chapter 7 we conclude and summarize the major points of the proposed framework as well as suggest future research and system expansion directions.

Chapter 2

Background and State of the art

The main research field of this thesis is Sponsored Search Advertising, a form of promotion that uses search engines with the purpose of suggesting marketing messages (specifically textual ads), with the goal of capturing the users' interests in order to interact with the ads and generate revenue. In this chapter, we introduce the background and the state of the art of sponsored search advertising.

2.1 Online Advertising

Online Advertising is an emerging research field, at the intersection of Information Retrieval, Machine Learning, Optimization, and Microeconomics. Its main goal is to choose the right ads to present to a user engaged in a given task. In general, advertising is a marketing mean that attracts potential customers to purchase a product or to subscribe to a service. In addition, it is a way to establish a brand image through repeated presence of an advertisement (ad) associated with the brand in the media. Traditionally, television, radio, newspaper, magazines, and billboards are among the major channels that place ads. The advancement of the Internet and the World Wide Web (WWW) enables users to seek information online. Using the Internet and the WWW, users are able to express their information requests, navigate specific websites and perform e-commerce transactions. Major search engines have been continuously improving their retrieval services and users' browsing experience by providing relevant results. The Internet and the WWW are therefore a natural choice for advertisers to widen their strategy in reaching potential customers among Web users. This phenomenon provides an opportunity for the search engine to be a strategic platform for advertisers to place their ads on the Web, with the view that a proportion of those who are online and seeking specific products or services may click the ads. Yuan et al. have observed in their study [106] that web advertising would possibly dominate existing media as the preferred medium for placing ads, because one of the major advantages that it has over traditional advertising media is that the former is more targeted. A user expressing his or her information need in the form of a query, e.g., *car rental*, is likely to respond to ads relevant to that query listed along with the organic search results. In comparison, ads in the newspaper have been pre-selected for display even before readers pick up their copies, and are less targeted and uniform for every reader. In addition, it is also not easy to measure the success of the advertising due to the lack of an effective feedback mechanism in the conventional media.

2.2 Sponsored Search

One of the primary channels for distributing ads is the paradigm of *Sponsored Search* (or Paid Search) Advertising. It was created in 1998 by Bill Gross of Idealab with the founding of Goto.com, which became Overture in October 2001, then acquired by Yahoo! in 2003 and is now Yahoo! Search Marketing. Meanwhile, Google started its own service AdWords using Generalized Second Price Auction (GSP) in February 2002 and added quality-based bidding in May 2002. In 2007, Yahoo! Search Marketing added quality-based bidding as well [27]. Web search has now become a necessary part of daily life, vastly reducing the difficulty and time that was once associated with satisfying an information need. Sponsored search allows advertisers to buy certain keywords to promote their business when users use such a search engine, and contributes greatly to its free service.

Sponsored search advertising is the task of displaying ads on the page returned from a Web search engine following a query. A commercial intermediary, namely ad network or auctioneer, is usually in charge of optimizing the selection of ads with the twofold goal of increasing revenue (shared between publisher and ad network) and improving user experience. Actually there are four main participants in the online advertising ecosystem as we can see from the study of Yuan et al. [106]: the auctioneer, the advertiser, the publisher, and the user.

An advertiser demands his ads to be displayed, whereas a publisher sells his ad inventories to gain revenue. In the case of sponsored search, a search engine acts as a publisher who has reserved space for ads on the search engine result page (SERP). An ad exchange is normally an advertising service providing the mechanism that enables advertisers to promote their products to targeted groups of users [77]. The ad network/exchange acts as auctioneer, selling keywords to advertisers. A major example of such an advertising service is Google AdWords. The match between a. {keywords, ads, and query terms}, b. {keywords and webpage contents}, and c. {keywords and user historical data} is processed by the auctioneer.

An advertiser requires spaces to place his marketing messages (i.e., ads) on search engine result pages in the sponsored search paradigm. Bid prices and the relevancies between bid phrases and user queries influence the awarded slot position. However, the bids placed by other advertisers on similar keywords are unknown, whether each bid will end up winning a slot is uncertain. Ads displayed at higher positions are more likely to be clicked on, therefore, advertisers typically compete to bid for keywords that they believe to be relevant to user queries to increase the chances that their ads will be placed at top positions.

A content publisher hosts websites that may reserve spaces for display advertisements. We note that a search engine acts as a publisher in the sponsored search case where conceptually its role is not much different from that of a content publisher.

A user issues ad-hoc topics to express his or her information needs. In organic search, the relevance between a search topic (query) and documents on the Web is used to retrieve relevant documents. However, in search-based advertising, ads are not retrieved purely based on relevance. The match between the search topics and the advertisers' keywords, the bid prices and CTRs for the keywords are among factors of deciding which ads are eventually given ad slots, although the exact method is unique from one search engine to another.

From the advertisers' point of view, the cost to advertise online is variable by choosing different pricing models, among which the most popular are pay-per-click (PPC), pay-per-mille (PPM), and pay-per-acquisition (PPA) with their corresponding costs, cost-per-click (CPC), cost-per-mille (CPM), and cost-per-acquisition (CPA).

In the PPM [31], the advertiser pays for every thousand impressions of their advertisement, thus it is considered unfair for the advertiser, because they cannot be guaranteed that their website will get the desired traffic (clicks). In the PPA [64], the advertiser pays only when the user performs an action (a conversion which usually is a purchase), thus it is considered unfair from the search-engine's point of view, since they might spend their advertising slots on websites that do not convert traffic into actions. The PPC [26] only charges the advertiser whenever an ad is clicked, which reflects the interest of the user. This is based on the effective targeting ability, which in turn leads back to the *best match* challenge. This model is considered to be the most fair for both the advertisers and the search engine provider. PPC works because it makes more sense for the search-engine to put an effort into allocating the correct advertising slots to those advertisers that will get more traffic. The advertisers also need to make more appealing websites so that they convert more, and make enough profit that at least covers the cost of advertisement. It also makes more sense for the advertisers to choose on which keywords they want to be advertised. This would not make sense in the PPA model, because if the cost per action is less than the profit margin of a product, then it does not matter on which keyword it is advertised, it is profitable anyway, but the search-engine is losing profits because the advertisement is not focused and therefore could produce more revenue. However, in the PPC model it matters if the keywords are relevant to the ads and to the websites being advertised, because then the users are much more likely to make a purchase (or convert clicks to actions in general).

Even if the pricing model is chosen, the final cost is variable due to the competition in ad auctions. These auctions are carried out every time the ads need to be displayed and also takes into account the quality score of historical performance and landing pages for the ads. This encourages advertisers to improve their campaigns in all aspects rather than increase the bids solely. We will discuss these concepts in more details in the following sections.

2.3 Definitions

In this section we describe important concepts and terminology related to sponsored search that will be discussed thoroughly in this thesis.

Search Engine Result Page (SERP) : It is the result page of any search query a web-user makes on a search engine. This is where advertisements (also known as sponsored results) are shown together with normal results (also known as organic results or algorithmic results).

Keyword : A word or phrase that can match a web-user's search query and at the same time describes the content advertised. An important note here it is that an advertising keyword is actually a keyphrase or else known as bid phrase and can contain more than one terms in its form.

Ad-text : The text that a web-user reads on an advertisement. It is known also as ad creative or ad copy.

Impression : The appearance of an advertisement in a SERP after a user's query

Click : The action of a web-user clicking on an advertisement with the result of being led to the advertiser's website.

Conversion : The action of performing a desired action (e.g., purchase, registration) after arriving to a website, following an advertisement.

Landing page : The specific webpage on the advertiser's website that the user lands on after clicking on an advertisement.

Bid : The maximum amount of money that an advertiser is willing to pay for a click on an advertisement that came from a specific keyword.

Cost per click (CPC) : The actual amount of money that an advertiser is being charged for a click on his advertisement. In most advertising platforms, we can see the following distinction. The average amount that an advertiser has been charged for a click on his ad is called *avgCPC* and it is calculated by dividing the total cost of his clicks by the total number of clicks. For example, if the ad receives two clicks, one costing $0.20 and one costing $0.40, the average CPC for those clicks is $0.30; avgCPC is based on

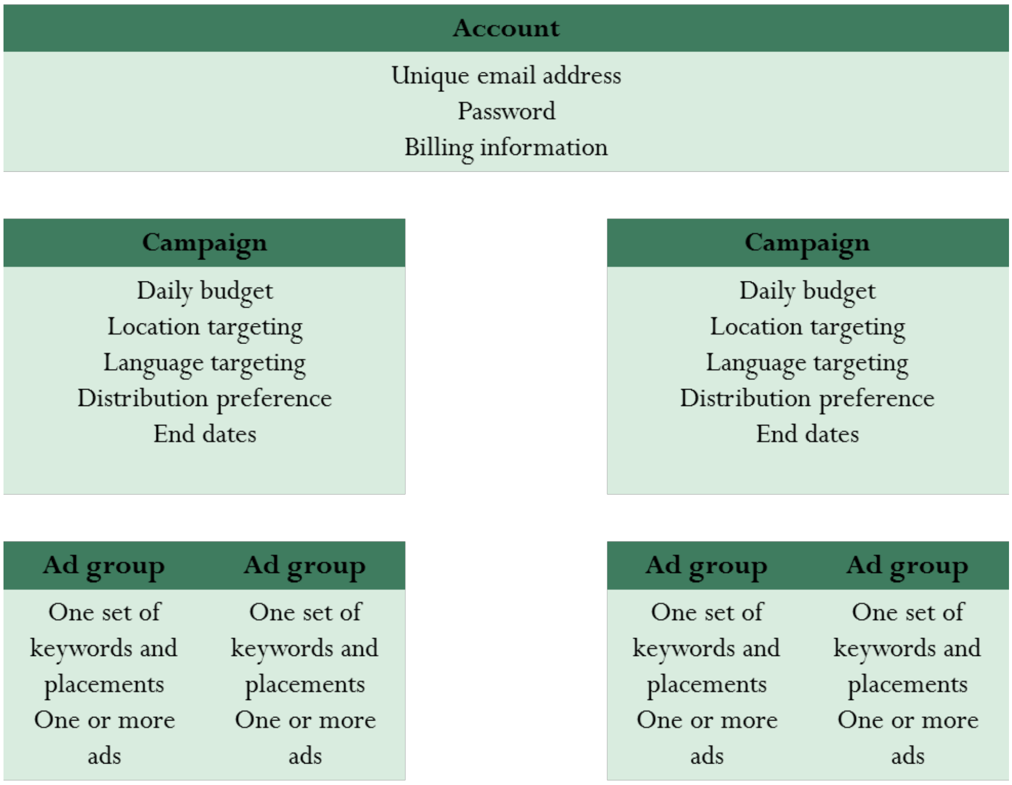


FIGURE 2.1: Hierarchy of the AdWords Model Organization

the actual CPCs (the actual amount an advertiser is charged for a click on his ad), which is different than the *maxCPC*; maxCPC is equivalent to the aforementioned concept of *bid*.

Click-Through Rate (CTR) : The percentage of people clicking on an advertisement when it appears in a SERP.
$CTR = Clicks/Impressions$

Conversion Rate (CR) : The percentage of conversions against clicks.
$CR = Conversions/Clicks$

Campaign : Set of general preferences and total budget for the advertising purpose.

Ad Group : Set of related keywords, ads, and bids within a campaign.

Figure 2.1 presents the hierarchy of the AdWords model organization along with basic characteristics of each entity.

2.4 Ad auction

If someone wants to advertise a product using search-based advertising there are a few things they need to do in order to participate in the *ad auction* process. First of all they need to choose the advertising platform where they will invest their money on.

Google Search is the search engine with the most searches world-wide. Google indexes all web pages regularly and ranks them by two factors: 1. word and phrase relevance (how many times the word appears in the page, where does it appear, synonyms that may appear, quality of the website), and 2. page rank (how many outside links point to that page determines how important it is) to finally produce relevant results to the users' searches. The normal results that Google shows based on ranking webpages are called organic results. Being advertised with Google Adwords means that your advertisements can appear along with the organic results in the search engine result pages (SERPs) of Google. These paid-for search results that complement original results are what Google makes its money out of, and are called sponsored results. They appear over the organic results (the first two or three sponsored results) and to the right side (the rest, to reach a maximum of 10 sponsored results per SERP), all having a slightly different background color than organic results. They also have a clear indication of being advertisements to distinguish them from organics ones, since people and companies are paying for them and they probably would not have made it as high on the results page, if they were not sponsoring. But being as high on the result page means better exposure to the public, so people are more likely to see the ads and get interested. This is crucial to companies of course, since they want to sell their products to as many people as possible and they are willing to increase the traffic on their website with potential customers by advertising on search engines. Figure 2.2 demonstrates the outcome of an ad auction. The ads are positioned above (3 highest ranked ads) and next to the organic results.

Google (and any search-engine) wants to maximize their profits, by keeping search-engine results quality in high standards. So performing a normal ad auction would not be sufficient. They have adopted a hybrid second-price auction [27] system that takes into account not only the bids of the advertisers, but also a parameter called *Quality Score*.

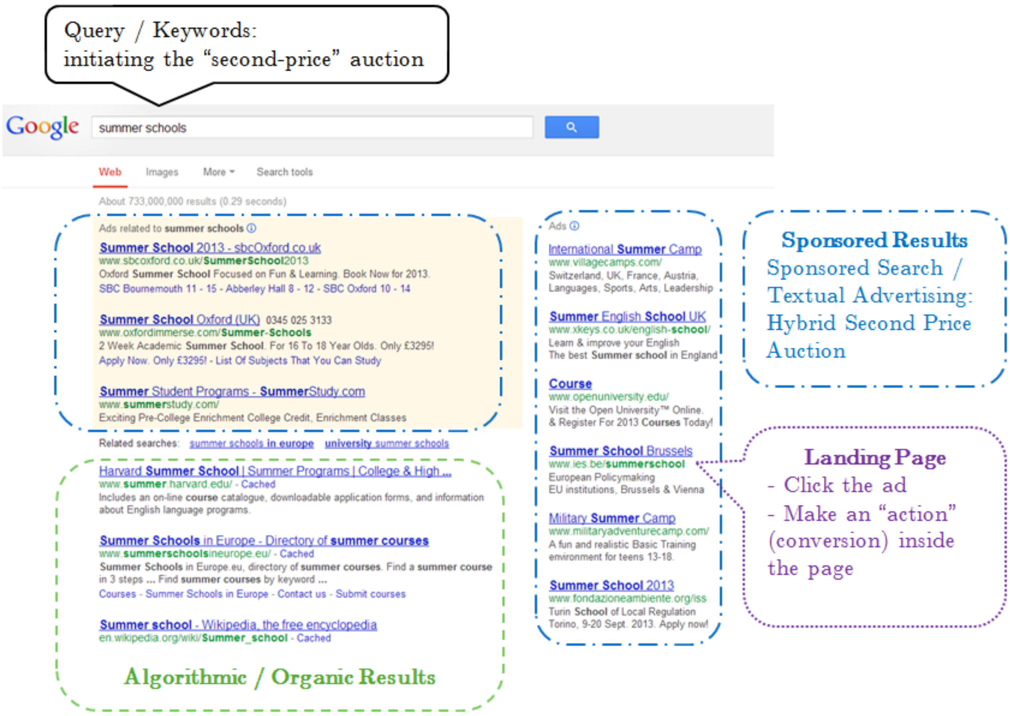


FIGURE 2.2: Outcome of an ad auction

We will present an example in order to demonstrate how a hybrid ad auction works. Alison, Bob, Cathy and Dave are four advertisers that want to bid on the same keyword competing for three available slots. In Table 2.1 is how a normal second price auction without quality score, would go (the advertiser with X is the one eliminated by the competition).

Advertiser	*Max Bid*	*Price paid*	*Position*
Alison	$4	**$3**	**1**
Bob	$3	**$2**	**2**
Cathy	$2	**$1**	**3**
Dave	$1	X	X

TABLE 2.1: Example of a normal second price auction

The more one bids, the higher the position they can get. Each one pays as much as the one below them. But Google’s quality score assignment makes the auction a bit more complicated. Quality score (QS) is an integer from 1 to 10 assigned to every advertiser that takes part to the auction, and is different according to the keyword. The higher the quality score, the better it is for the advertiser. In Table 2.2, the four advertisers have different quality scores, and their position changes according to their ad rank which is the bid multiplied by the quality score.

Advertiser	*Max Bid*		*QS*		*Ad rank*		*Position*
Alison	$4	*	1	=	4	→	X
Bob	$3	*	3	=	**9**	→	**2**
Cathy	$2	*	6	=	**12**	→	**1**
Dave	$1	*	8	=	**8**	→	**3**

TABLE 2.2: How bidding affects positioning

In the end Cathy got the first position because she had the highest ad rank, and Alison was eliminated even though she was willing to pay more than anyone else. The price that is actually paid by the advertisers for each click is given by dividing the ad rank of the one beneath him with his quality score.

$$P1 = (B2 * Q2)/Q1$$

where P=Price, B=Bid, Q=Quality score.

Table 2.3 shows how the final prices of each advertiser will form.

Advertiser	*Position*	*Max Bid*	*QS*	*Price Paid*
Cathy	1	$2	6	**$1.5**
Bob	2	$3	3	**$2.6**
Dave	3	$1	8	**$0.5**
Alison	X	$4	1	X

TABLE 2.3: How is pricing calculated

What is more interesting is how Google assigns the quality score to each advertiser for each keyword. According to Google [40], quality score is based on 3 components:

1. Click-through rate (60%): Each click on an advertisement from a user is considered as a vote of quality, in terms of keyword relevancy and ad quality. It is not only the click-through rates of keywords that matter, but also that of the ad-groups and campaigns in which they belong to.

2. Keyword relevancy (30%): If an advertiser bids on a keyword, this keyword has to be as relevant as possible to the displayed ad and the search query of the user.

3. Landing page quality (10%): The landing page of the advertisement must be in the same context as the keyword and have original content. Moreover, it helps to have good qualities, such as quick page load times and good page navigation. This way Google keeps a good balance between maximizing profits and keeping search results relevant to the users' queries. The advertisers are forced to bid only on relevant keywords and at the same time have good-quality landing pages. Competition works best for all parties involved. Looking for good keyword opportunities is probably best way to get profitable advertising campaigns.

2.5 Developing a sponsored search advertising campaign

To create an advertising campaign usually the advertiser must have one or more products on their website that he/she wants to be exposed to the public. For each product there must be a landing page, which is the web-page a user will land on, after clicking the advertisement of the product. The landing page is usually the place where the user can see information about the product, its technical characteristics, its price, and has the option to buy it.

After finding what the advertising wants to sell and preparing the landing pages, it is crucial to select on which keywords (words or phrases) each product will be advertised. The keywords used for each product must be relevant to it otherwise the campaign will not be profitable. A good practice is to choose the most specific key-phrases possible, which usually consist of many words. This is true for two reasons: firstly because long specific phrases usually have less competition, thus are expected to cost less, and secondly, because when a user searches for something more specific they are more likely to convert.

After finding the keywords, the advertisement texts must be correctly written. They must be short and precise, understandable with convincing calls to the user to take action. Ad-texts consist of a short headline, two limited lines of description and a display url, which does not have to be the same as the real landing page url.

Keywords and ad-texts belong to adgroups. The ad-texts of an adgroup are shown for keywords belonging to the same adgroup. So it is important not to mix ad-texts and keywords of irrelevant products in the same adgroup.

A budget must also be set on every campaign which will be consumed by keywords of its adgroups. An advertiser must also decide how much the maximum cost-per-click of each keyword will be. This is the bid that the advertiser is putting for a keyword and approximately an upper limit of how much each click for this keyword may cost.

After bidding on keywords one can see a lot of useful information and statistics that we have decribed earlier about the performance of these keywords such as impressions, clicks, CTR, average CPC, quality score, total cost, conversion rate, average position, and a rough estimation of how much each keyword would cost to appear in the first page of results. All these are valuable information on the performance of each keyword. These statistics are available only after a period of time where keywords along with their bids have been tried, so there is no way to know a keyword's CTR beforehand. Different bids on a keyword result to different keyword behavior. If a keyword reaches a high position, it is likely to have higher CTR but also higher average CPC, so a balance must be achieved. For a limited budget, an advertiser has to decide for each keyword if it is worth the investment and if it is, how much should it cost. Because in the end, some keywords are more cost-effective than others and the maximizing of profits comes only from *selecting the best ones*. This is a demanding task even for PPC advertising specialists, especially if one has to choose among hundreds or thousands of options.

Chapter 3

Keyword Generation

In this chapter we describe our approach regarding the *keyword generation task for online advertising* [95, 97] which aims at proposing valid and representative keywords for a landing page capitalizing on keyword extraction methods, on the co-occurrence of terms, and on keyword suggestions extracted from relevant search result snippets.

3.1 Introduction

Keyword selection [92] is a technique that is used within Search Engine Marketing (SEM) in order to attract traffic to a web site and get potential customers. This technique, when used for sponsored search advertising purposes, is also known as *bid phrase generation* [14, 83], *keyword extraction and suggestion* [53, 60, 87, 105, 110], keyword research [70], or *keyword generation* [1, 50]. In fact, the goal of the task is to find the proper phrases for corresponding promotion, so the *advertising keywords* here are not only a single word but can be multiword terms[1].

[1]Henceforth, when we are referring to keywords, we mean the advertising definition of them, thus the more precise concept is that of *keyphrases*

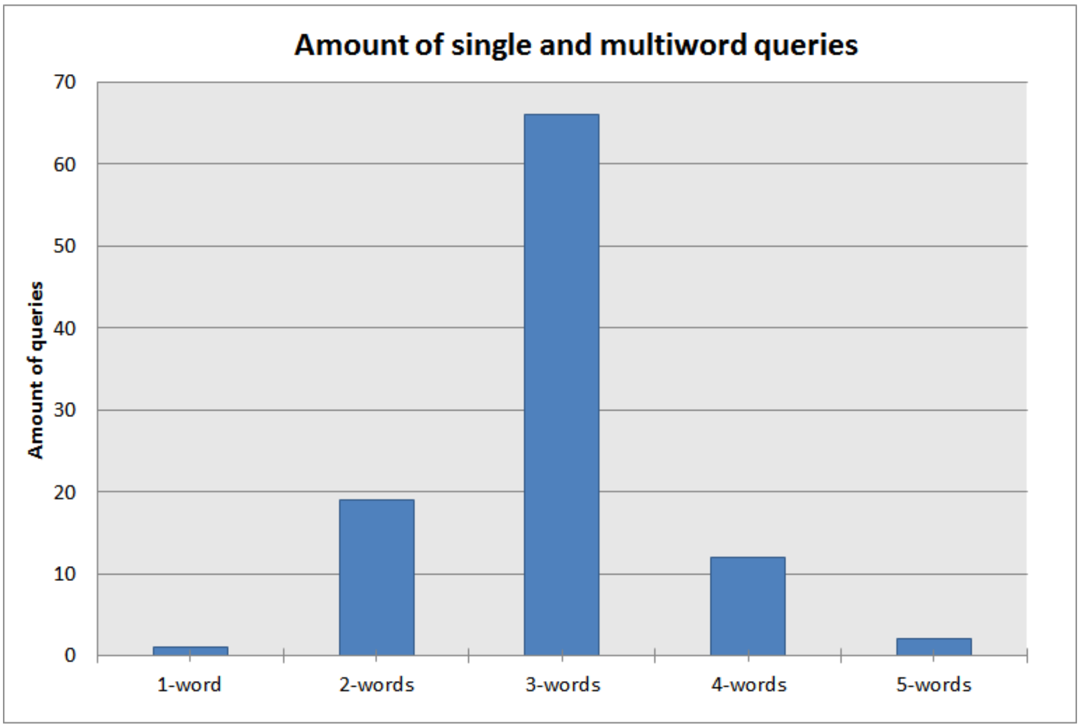


FIGURE 3.1: Frequency for single and multiword queries

3.1.1 Important Properties

Keywords can attract different types of users, and these different groups have a different purpose and a different conversion ratio. For example, people that use a phrase (three or more search words) as a query, are searching in a more *specific* way, meaning that when they enter finally the landing page the chance of conversion is higher than those users that have searched using a single, very popular keyword. Figure 3.1 shows stats that are retrieved from Google Analytics [2] and present the distribution between single and multiword form of queries for the top-100 searches for a car rental website.

Some keywords are used by many competitor advertisers and are therefore popular (e.g., "seo") , a fact that causes these terms to be expensive and generate traffic with a low conversion rate since only a few people who search for it will actually buy the product. One can also say that this term is too obvious. Thus, another interesting concept that arises is the *nonobviousness* factor. There are existing different notions for this concept. In [88] a nonobvious keyword is a keyword that is not so popular and thus *cheaper*; consider the quantitative competition between "seo" and "corporate reputation mining". The latter

[2] www.google.com/analytics/

phrase might not generate a lot of traffic, however this traffic that will eventually generate could have higher conversion rate due to its more specific nature. Because these long-tail keywords are cheap, a lot of those can outperform a single popular keyword, for equal or less money. In [50] the authors defined a nonobvious term as a term not containing the seed keyword or its variants sharing a common stem. In this thesis, we define nonobviousness more close to the concept of overall variety inside the final keyword set and not as measure for a single keyword. This decision came from the fact that we want to suggest to the advertiser an interesting group of phrases and extra terms that are not readily apparent.

The next important aspect is *relevance*. Attracting traffic with a high conversion rate requires relevant keywords, i.e., keywords that match contextually as well as possible with the contents of the website or the specific product that a landing page offers. Besides, quality score is heavily based, as we have described in Chapter 2, on the relevance of keywords and the promoted landing page.

Concluding, the goal is to find the right balance between specific, nonobvious, cheap, and relevant keywords.

3.1.2 Term Weighting Background

In a web page structure, the text fields represent the semantics of the page. According to the vector space model, each web page can be considered as a document and its text content must be segmented as many weighted keywords which all together represent the semantics of a document [110]. After segmentation of text, the result will be a set of keywords usually called a "term" in a document. Then each of these terms must be weighted properly to assure that terms with higher semantic meaning and relevance to our page have larger weight. Towards this end, we need to assign to each term in a document a *weight* for that term, that depends on the number of occurrences of the term in the document. We would like to compute a score between a query term t and a document d, based on the weight of t in d. The simplest approach is to assign the weight to be equal to the number of occurrences of term t in document d. This weighting scheme is referred to as *term frequency (tf)* [68] and can be computed as:

$$tf_{t,d} = \frac{f(t,d)}{\sum_{j=1}^{n_d} f(t_j,d)} \tag{3.1}$$

where $f(t,d)$ is the term frequency of t in d and n_d is the number of distinct terms in d. tf captures the importance of a term t in a document by assuming that the higher tf score of t, the more importance of t with respect to d. Intuitively, terms that convey key concepts of a document should have high values of tf.

idf is the inverse document frequency weight of a term t. It measures the importance of t with respect to a document collection. Denoting the total number of documents in a collection by N and document frequency of the t (number of documents in the collection that contain the term t) as df, we define idf of a term t as follows:

$$idf_t = log\frac{N}{df_t} \tag{3.2}$$

Leveraging this measure, the weight of given term is calculated in the following equation called *tf-idf* scheme after all the documents are processed:

$$tfidf_{t,d} = tf_{t,d} \times idf_t \tag{3.3}$$

The reason that makes *tf-idf* scheme useful comes from the fact that uses a mechanism for attenuating the effect of terms that occur too often in a document collection to be meaningful for relevance determination. This is an important technique in information retrieval and for corpus dependent keyphrase extraction approaches.

Another interesting weighting function is that of *Okapi BM25* [84] and more specifically the *BM25F* variant [108]. The authors introduced this variation for structured documents, i.e., consisting of different fields. These fields were initially referring to title and body. But with information retrieval over HTML documents, BM25F became a useful tool, as web page may consist of a number of fields of different importance (e.g., title, keywords, description, headings, etc.). BM25F basically introduces pseudo-frequencies in the BM25 ranking function, where different fields have different weights. The field-dependent term frequency is computed as the tf component in the following equation but in the scope of the corresponding field. Then, as a term may occur in more than one field, these frequencies are aggregated:

$$tf_{t,d} = \sum_{field} w_{field} \times tf_{d,t,field} \tag{3.4}$$

3.2 Related Work

In this section, we provide an overview of related work. Since keyword suggestion is practiced by each of the parties namely the search engine, the advertiser and the searcher, the forms of input used to suggest keywords and the type of keywords generated also differ. On each sort of input there are also different techniques identified to process the input and generate the keywords. In our thesis we tackle the problem of keyword generation for the assistance of the advertiser role, thus we studied the relevant research literature and commercial from this perspective. In the following paragraphs we analyse the different approaches.

Query-click and advertiser log mining. Search engines use query log based mining tools to generate keyword suggestions. In this way, they focus on discovering co-occurrence relationships between terms and suggest similar keywords. They start from an initial key phrase and they are based on past queries that contain these search terms. Google Keyword Suggestion Tool exploits this ability and presents frequent queries for the seed set of terms. In addition with this approach, Google as an auctioneer can exploit advertiser log mining. This approach makes use of words used by advertisers and exploits co-occurrence relationships in advertiser logs. Other commercial systems determine an advertiser's top competitors and then actively search for the keywords they are targeting. After a period of time, lists of targeted keywords that are competitive for pay per click advertising are automatically generated. Keywords generated by taking into consideration traffic reports are limited to words that occur frequently in advertisers search logs. Also, they are likely to be expensive because of their competitive nature among a large amount of advertisers. Furthermore, in several cases they are not so relevant, because these techniques favor more general terms and not specific keywords that the advertiser would ideally choose to match his text ad. Consequently, this approach results to an increase in the bids for the suggested keywords to compete efficiently for a good rank among appearing text ads. However, increasing the bid to win a high position in the sponsored results does not guarantee profit increase because general terms result to high clickthrough rates but in the same time to low conversion rates.

Public taxonomy and concept hierarchy. Chen et al. [18] introduce the use of taxonomies available on the web (like DMOZ[3]) to build a tree of concepts. A concept is in this case a keyword, placed in an hierarchical tree. Using this tree one can find related concepts and use them for suggestion. This technique result in a concept graph which is used to suggest related keywords based on a seed keyword. The disadvantage of the concept based approach is that it is not flexible enough, since the concept hierarchy is a static structure and does not catch up with the changing world, unless it is rebuild.

Domain dependent system. In a similar manner, there are existing keyphrase extraction methods for alternative purposes, not focused on the advertising tasks, such as KEA [102], which builds a Naive Bayes model with known key phrases, and then uses the model to find keyphrases in new documents. KEA requires a training step on a number of documents, thereby imposing a burden on potential users. The required training step, where at least 20 documents with assigned keywords have to be used as training collection in order to obtain useful results increases the burden for out-of-the-box application by the ordinary user. S. Ravi et al. [83] propose a generative model within a machine translation framework so the system translates any given landing page into relevant bid phrases. They first construct a parallel corpus from a given set of bid phrases b, aligned to landing page keywords l, and then learn the translation model to estimate $Pr(l|b)$ for unseen (b, l) pairs. This approach performs efficiently but depends on the chosen domain and data that the human decision factor may affect. In general, *corpus or domain dependent* systems require a large stack of documents and predetermined keywords to build a prediction model [60], while on the other hand our proposed approach, that we will present in the next section, works with a *corpus independent* approach that directly extracts

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

keywords from a single document without any previous or background information, expanding them also with knowledge from snippets of search engine results.

Web-based feedback. TermsNet and Wordy [1, 50] exploit the power of search engines to generate a portfolio of terms and to establish the relevance between them. After selecting the most salient terms of the advertiser's web page they query search engines with each initial seed term. With their methods they find other semantically similar terms. The Wordy system proposed single word terms (unigrams) for each seed keyword. This approach is similar with our proposed method for extra keyword suggestions, while we are generating multiword terms (bigrams, trigrams).

3.3 Keyword Extraction

The *Keyword Generation* task that we propose begins with the subtask of *Keyword Extraction*. In this process, we follow the *corpus independent* approach to rely solely on the given landing page document. Due to the form of the examined document in our task, which in our case is a single landing web page (a single HTML document) and the *corpus independent* approach, we do not have the idf_t parameter. Instead, we apply a *single document keyword extraction* [72] method, which could get weighted keywords based on single document using word co-occurrence and field information.

As a preprocessing step, the HTML content of each landing page is parsed, stopwords are removed and the text content is tokenized.

Next, for each word (gram) j in the tokenized output, we compute a weight associated with the gram for each occurrence inside a specific tag, e.g. the occurrence of a gram inside $< h1 >$ tags.

$$w_{j,tag} = weight_{tag} * f_{j,tag} \tag{3.5}$$

where $weight_{tag}$ is a special weight assigned to each different HTML tag and $f_{j,tag}$ is the frequency of the gram inside the specified tag. The weight of each tag is assigned according to its importance inside the HTML document. We set higher values on important tags such as $< title >$, meta keywords, meta description, anchor text, $< h1 >$, $< b >$. In Table 3.1 we propose the assignment of tag weights following an approach that ranks the importance of these tags according to where web page designers choose to place the most important information on their website. These weights are proposed after parameter tuning, evaluating, and verifying that the best n-grams were returned along with a proper order.

Then, we compute the special weight of each gram as the sum of all $w_{j,tag}$ weights:

$$special\_weight_j = \sum w_{j,tag} \tag{3.6}$$

In the next step, the relevance score of each gram is computed:

$$relevance\_score_j = \frac{special\_weight_j}{MAX\_WEIGHT} \tag{3.7}$$

where MAX_WEIGHT represents the maximum special weight that a gram could have inside the HTML document. MAX_WEIGHT can be different for each HTML document because some of them may not have links or bold tags, etc. Thus, the MAX_WEIGHT is simply the sum of all the tag weights of the specific page.

Unimportant unigrams occurring on the page are filtered out using a threshold $\tau = 0.001 * relevance_{max}$, resulting after several experiments and parameter tuning on the relevance score. While unigrams frequently have a broad meaning,

Element	**Assigned Weight**
<title>	50
meta keywords	40
meta description	40
anchor text	30
<h1>	30
<b>	10
other	1

TABLE 3.1: Tag Weights

multiword phrases (n-grams) are more specific and thus can be more representative as advertising keywords. A typical query length, especially while searching for a product or service, varies between 1 and 3 grams. For that reason, from the extracted single word terms (unigrams) we pull together possible combinations of two-word phrases (bigrams) inside the given landing page. Next, in order to construct the gram co-occurrence matrix, the top N grams with high relevance scores are ranked in descending order. Then we define co-occurrence as follows: if $gram_i$ and $gram_j$ appear in a same unit (each different area inside an HTML document, defined by HTML tags) which is predefined, then they co-occur once, and $freq_{i,j}$ should be increased by one.

Finally, we consider the co-occurring two-word terms (bigrams) above τ and follow the same process, searching for new co-occurrence with each unique unigram. In this way, we extract three-word terms (trigrams) as well. By gathering all terms, we construct the extracted keywords vector. In order to boost trigrams first, bigrams second and unigrams third, we modify their relevance score with the following factor:

$$boosted\_score_j = relevance\_score_j * k^{noOfGrams} \tag{3.8}$$

where k is a free parameter (in our experiments we set it to ($k = 100$) and $noOfGrams$ is the number of grams composing a term.

3.4 Keyword Suggestion

From the previous step of keyword extraction we have already extracted the initial keywords. These will be the seed keywords for the additional suggestions. Initially, as this procedure begins, the set of additional suggestions is empty. We provide as input the extracted keywords from the landing page. For each given seed keyword, the keyword is submitted as a query q into a search engine API. We use for this purpose Google JSON/Atom Custom Search API [4]. With this API, developers can use RESTful requests to get search results in either JSON or Atom format. The API returns a set of short text snippets, snippets that are relevant to the query and thus to the keyword. In this way, what we aim

[4] http://code.google.com/apis/customsearch/v1/overview.html

to achieve is the discovery of new nonobvious phrases that are not appearing inside the landing page.

From the response data we retrieve `/entry/summary/text()` which is a string type property indicating the snippet of the search result and `feed/entry/title/text()` which is a string type property indicating the title of the search result. The top 30 results are downloaded and loaded in Apache Lucene Library [5], which we use for implementing indexing and query support in our system. Each extracted term from the previous step which was a seed for the query has now been extended to a set of results which we use as a document in the Lucene index. Each set of title and snippet results that were retrieved after a seed query represents this document d for Lucene indexing.

In this step, we parse the resulting document and construct a new vector of grams $< g_1, g_2, \ldots, g_{|d|} >$. Based on the Lucene scoring method we find the unigrams and bigrams that have the largest number of occurrences inside the document and thus are kept as the most relevant for the specific seed query. Each of these terms is representing a new query q'.

The score of query q' for document d is considered to be the cosine-distance or dot-product between document and query vectors in a Vector Space Model (VSM) of Information Retrieval. Again, we sort in descent order the new queries based on this score and we create a vector of suggested keywords and their scores for each of the seed terms. Before we place our output as an integrated keyword set to the advertiser, we normalize scores from Keyword Extraction and Keyword Suggestion processes using min-max normalization:

$$relevance' = \frac{relevance_i - relevance_{min}}{relevance_{max} - relevance_{min}} \tag{3.9}$$

where $relevance_{max} - relevance_{min} \neq 0$.

Finally, we use a new threshold ($\tau = 0.5$ which represents the 50% of the maximum relevance) for keeping only the most salient terms.

[5] http://lucene.apache.org/java/docs/index.html

3.5 Evaluation

In order to evaluate the performance of the proposed keyword generation system for online advertising purposes, we carried out a qualitative evaluation in order to assess the potential utility of our prototype compared to existing solutions that are being used already by the majority of the advertisers. As a next and more straightforward evaluation, we proceeded with a real-world campaign experiment focused on CTR performance.

3.5.1 Comparison with State-of-the-art commercial systems

We evaluated at a first glance our method using human ranking for selected keywords following a blind testing protocol. The landing pages for our experiments were taken from different thematic areas, promoting several products and services. The categories were:

1. hardware product
2. corporate web presence optimization service
3. gifts
4. GPS review
5. hair products
6. vacation packages
7. web design templates
8. car rental services

To compare our system results, we used the following competitive keyword selection systems:

1. GrammAds (our prototype)
2. Google Keyword Suggestion Tool [6]

[6]Most recently known as Google AdWords Keyword Planner https://adwords.google.com/KeywordPlanner

3. Alchemy API [7]

4. Google AdWords API RelatedToUrl method [8]

We constructed a dataset of each method's selected keywords in order to start a blind experiment evaluation. Eleven researchers and informatics postgraduate students provided judgments for each system output regarding keyword relevance, specificity and nonobviousness using a scale of 1 (Very Poor) to 5 (Very Good).

For each of the 8 landing pages, we applied the 4 keyword generation systems. We kept approximately [9] the top-20 results from each system as a suggested set, resulting to 32 keyword sets. Each judge had to evaluate approximately 3 (different) sets. Each judge knew only the correspondence of keyword set and landing page.

Test measures and evaluation guidelines were defined as follows:

Relevance : The relevance of keywords related to each landing page. Score(1) - None of the keywords in the set are relevant with the promoted product or service of the landing page; Score(5) - Very good overall set relevance.

Specificity : How general or specific were the generated keywords. Score(1) - None of the keywords in the set are focused on the specific promoted product; Score(5) - Very good overall set specificity.

Nonobviousness : Overall variety of combinations on the final set; How redundant and repeatable or nonobvious and diverse was the finale set related to the category and advertising form of each landing page. Score(1) - This set is a complete repetition of the same terms; Score(5) - Very good overall set variety and interesting suggestions.

In order to present more thoroughly to the reader the meaning of the *nonobviousness* measure, we provide as an example Table 3.2. The first and poorly

[7]http://www.alchemyapi.com/

[8]Known as RelatedToUrlSearchParameter for the API Version of May 2014 https://developers.google.com/adwords/api/docs/reference/v201402/TargetingIdeaService.RelatedToUrlSearchParameter

[9]This approximation is a result of the inability in some cases to achieve more than 10 results; this was an issue especially with the Google AdWords API RelatedToUrl method

performed snapshot is a result from the Google AdWords API RelatedToUrl method. This method cannot return a lot of keywords and diverse combinations due to API regional and other limitations for the landing pages. The second snapshot is a well-performed result of our GrammAds prototype.

Nonobviousness	Keyword set snapshot for a SEO company landing page
Score(1): Complete repetition of the same terms	seo seo company seo optimization seo optimization company
Score(5): Very good variety of terms	search engine optimization corporate reputation mining text mining services search engine marketing

TABLE 3.2: Example of nonobviousness fulfilment

Tables 3.3 presents a snapshot of the generated keywords by each system for a corporate web presence optimization service. The results came directly by placing the corresponding landing page as a url input to each system, thus we provide here the exact outcome that contains some phrases from a footer section that had not been preprocessed properly by the systems.

Figures 3.2 - 3.4 and table 3.4 show the relevance, specificity, nonobviousness, and overall scores of the 4 systems. All scores were normalized in [0,1] (we assume as a criterion fulfilment scores 4 and 5, while sets that have been scored as 1, 2, and 3 are considered as mediocre and poorly performed). As an overall score we calculate the **Harmonic Mean**, as we need an average of rates for the previous criteria.

Our GrammAds prototype outperforms overall the Alchemy and AdWords RelatedToURL methods. In addition, it outperforms Google Keyword Suggestion Tool in terms of nonobviousness. GrammAds does not outperform Google Keyword Suggestion Tool overall, but it seems to perform well against it. Google Keyword Suggestion Tool can achieve this performance due to its vastly large resources of similar landing pages, query logs, and complete campaign data as Google is the auctioneer of the advertising process. An explicit comparison at this stage with Google Keyword Suggestion Tool would be unfair. As a remark

GrammAds	*Google Keyword Suggestion*	*Alchemy*	*Adwords RelatedToURL*
text mining services	search engine placement	competent technical staff	seo company
web services personnel	search engine optimization seo	search engine optimization	what is seo
web pages services	seo optimization company	world class researchers	seo optimization company
corporate reputation mining	seo company	online advertising services	search engine placement
data mining reputation	seo companies	Our team	
web advertising development	website optimization company	competitive international context	
search engine marketing	search engine placement companies	efficient solutions	
search engine optimization	seo optimization services	internet marketing	
company vision personnel	best seo company	art research	
vision personnel services	what is seo	innovative	
web pages design	seo search engine	advertising	
news contact change	search engine placement company	engine	
introducing mining	search engine optimization firms	corporation	
marketing search	local search marketing	image	
mining services	local search engine optimization	international	
web services	google search optimization		
reputation mining	website optimization services		
corporate reputation	top seo companies		
web development	local search engine marketing		
web advertising	small business seo		

TABLE 3.3: Results for a sample landing page

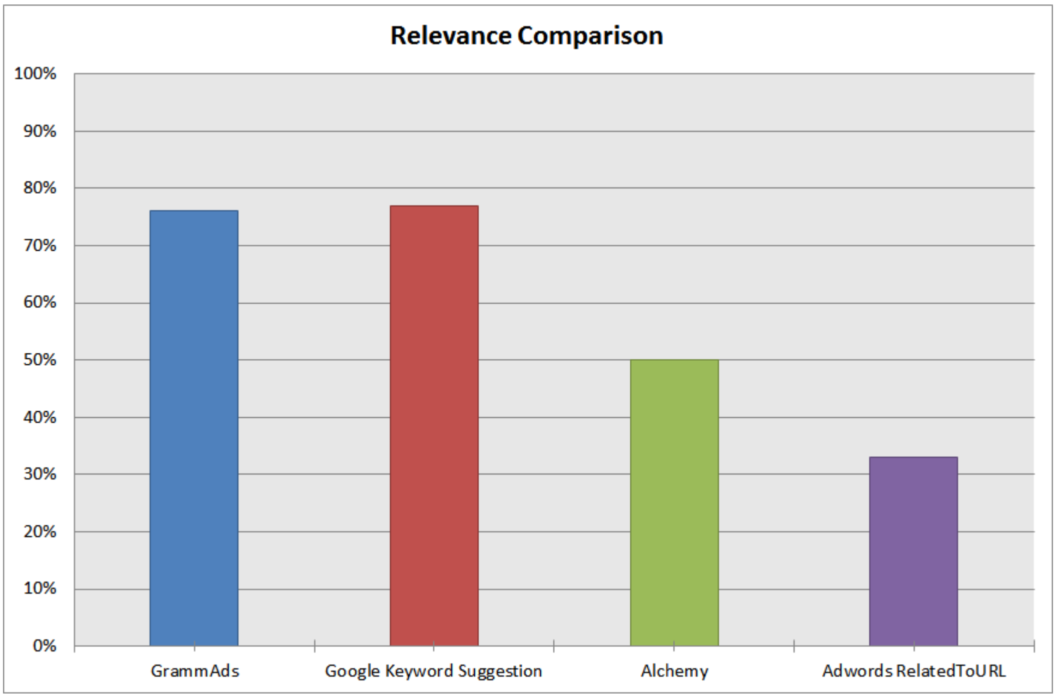


FIGURE 3.2: Relevance comparison with competitive keyword selection systems

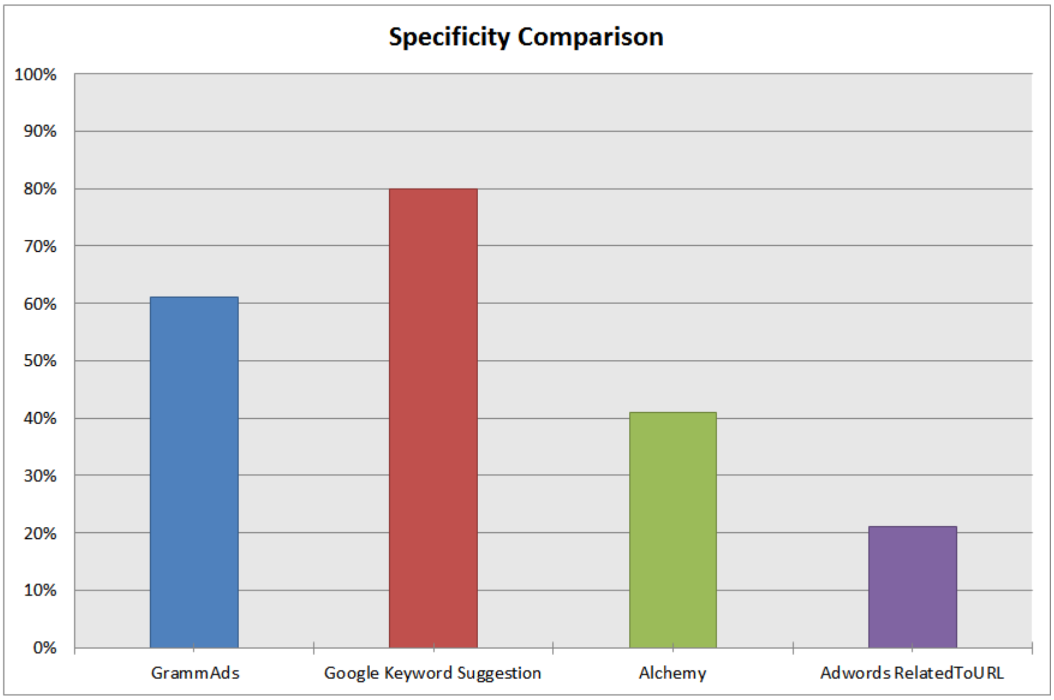


FIGURE 3.3: Specificity comparison with competitive keyword selection systems

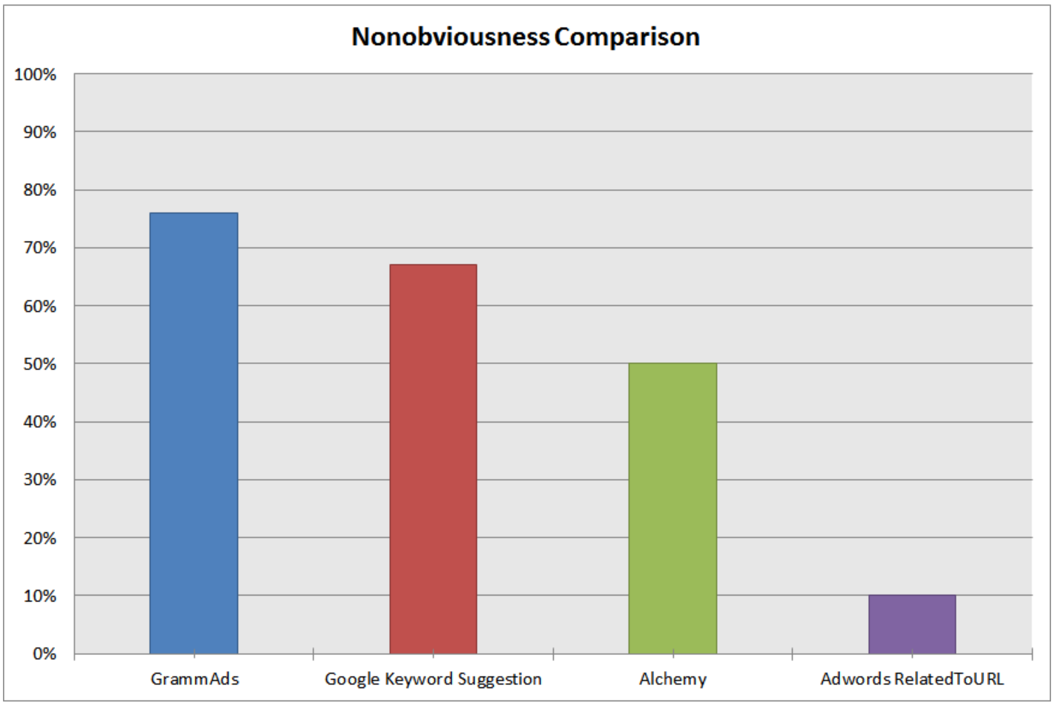


FIGURE 3.4: Nonobviousness comparison with competitive keyword selection systems

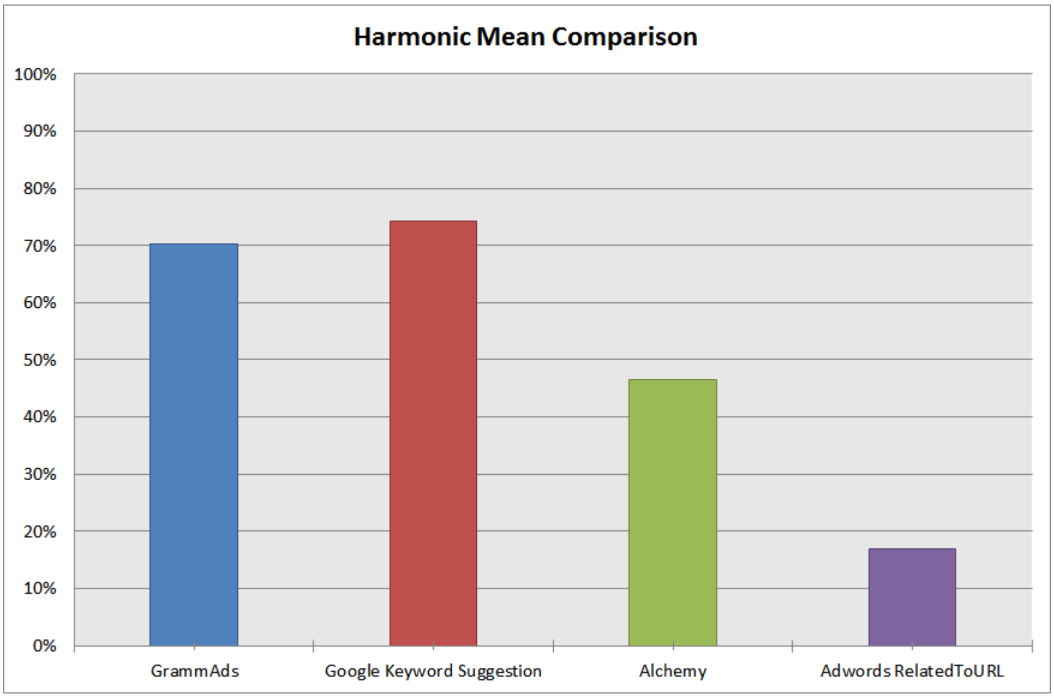


FIGURE 3.5: Harmonic Mean comparison with competitive keyword selection systems

	GrammAds	Google Keyword Suggestion	Alchemy	Adwords RelatedToURL
Relevance	0.76	0.77	0.50	0.33
Specificity	0.61	0.80	0.41	0.21
Nonobviousness	0.76	0.67	0.50	0.10
Harmonic Mean	0.70	0.74	0.46	0.17

TABLE 3.4: Summarizing results of comparison with competitive keyword selection systems

on this, Google Keyword Suggestion Tool leads in many cases to very expensive keyword suggestions for bidding, thus our prototype could be considered as a very good performing alternative.

3.5.2 Real-world Campaign Evaluation

A more precise indicator of the proposed method performance is to actually carry out an experiment with a real-world campaign and measure the CTR performance of the selected keywords. The choice of CTR as a way of measuring the success relies on the purpose of convincing a user to click when he has already saw a result. This means that the advertiser captured customers' response to a landing page because at this point the user perceived the result either as very relevant to his query or very interesting in general.

In order to conduct a comparison regarding only the keyword set, we used the exact same bidding strategy and ads for two identical campaigns of a prefabricated housing company. The campaigns were active for a period of 2 weeks[10]. For the first campaign we used as keywords a provided set with human selected phrases and we name these as "manual options". For the second campaign we used the resulted keyword set from our GrammAds prototype. We let as bidding strategy for both of the campaigns the automatic administration by the corresponding tool of AdWords. This tool assigns some initial bidding values to each keyword and then in regular intervals does a simple check for 0% CTR in order to pause some keyword options in the next day.

[10]We thank **Google Greece** for providing us coupons in order to run the experiments. The average budget that we used in this experiment and in those of the following chapters was **100 euros**

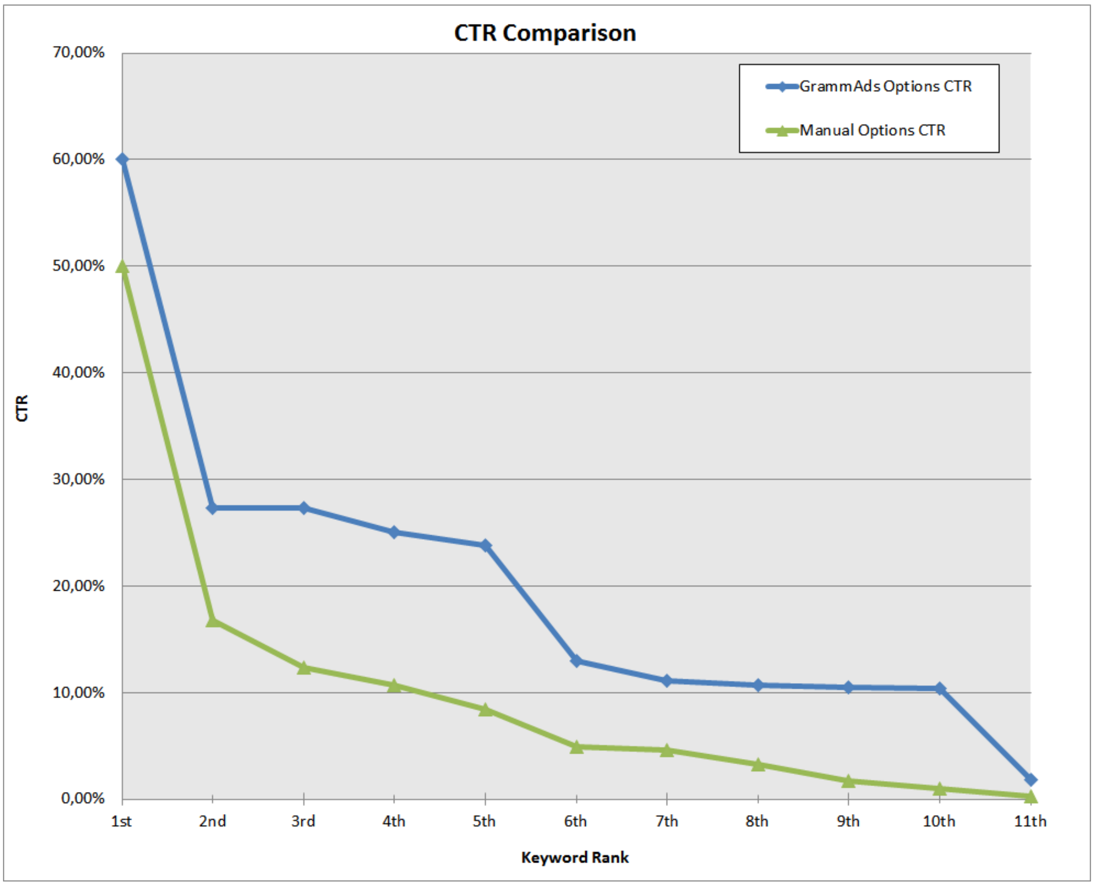


FIGURE 3.6: Keyword CTR Comparison for top-11 terms

The keywords that were generated from GrammAds achieved higher Click-through rate (CTR) values than the manual inserted ones as shown in Figure 3.6. The comparison is between the top-11 keywords in terms of CTR performance from each method (GrammAds Options vs. Manual Options). We observe also a faster drop in the CTR rates of the manual options. A necessary remark here is that in general an average CTR campaign rate of 2% is considered a usual result [56], but in this experiment we do not measure the overall campaign performance but rather the individual keyword rates, thus this is why such high CTR values are observed.

Chapter 4

Ad-Text Generation

Products, services or brands can be advertised alongside the search results in major search engines, while recently smaller displays on devices like tablets and smartphones have imposed the need for smaller ad texts.

In this chapter we tackle the novel issue of *automatic ad-text generation*. We propose a method [98] that produces in an automated manner compact text ads (promotional text snippets), given as input a product description webpage (landing page). The challenge is to produce a small comprehensive ad while maintaining at the same time relevance, clarity, and attractiveness.

4.1 Introduction

The *ad-text generation* task aims at facilitating the process of online advertising. The main notion is to provide an efficient solution for online marketing campaigns that feature large websites or shops that can be considered as large online product catalogues. These sites may include hundreds or thousands products or services that each one of them need to be promoted through a text ad. At the same time, there is an emerging need for promotion through channels that require more and more short promotional text like interfaces on tablets and smartphones. In this way, our proposed method contributes with the automated generation of compact but comprehensive ad text.

On the surface, our ad-text generation task appears to be similar to a keyphrase extraction problem. However, since the goal is to help users digest the underlying promoted messages through the ad snippets, there are some important aspects that are unique to this task. In traditional keyphrase extraction, the goal is primarily to select a set of phrases to characterize documents. In the task described here, we want users to understand the text in the snippet, thus it is more similar to a landing page summarization task. The phrases in the generated summary need to be fairly well-formed and grammatically sound. Consider a phrase such as "effective refresh rate" in contrast to one such as "rate effective refresh". Even though both phrases contain the same words, the ordering is different, changing their meaning, where the former is readable and the latter is not. This readability aspect is less of a concern in keyphrase extraction tasks as the phrases are only used to "tag" documents.

Another important aspect is the following. The vast majority of the current websites that refer to a large available set of products or services (e.g., Amazon, eBay, etc.) offer the possibility for a client to write a *review* or a *comment* for any purchased product. Therefore opinion mining may play an important role in our approach by separating the negative phrases from the positive ones, providing a set of phrases that could potentially be used for marketing purposes. Thus, we want to leverage for information feedback not only description fields that contain the features of a product but other interesting segments as well, such as positive reviews.

4.1.1 Important Properties

The integral parts that compose a Google AdWords text ad are the following:

1. Headline of the textAd $head$: The problem or opportunity; Ad titles are limited to 25 characters
2. Description Line 1 dl_1: Short description of big benefit; limited to 35 characters
3. Description Line 2 dl_2: Short description of the product/service; limited to 35 characters

4. DisplayURL $url_{display}$: The web site's URL up to 35 characters; Google can only display up to 35 characters of the display URL, due to limited space. If the display URL is longer than 35 characters, it will appear shortened when the ad is displayed

5. DestinationURL $url_{destination}$: Landing page: URL of the exact Web page customers visit first

Henceforth, when we are referring to the promoted snippet, this corresponds to the two description lines of 70 characters in total.

Ad creatives often require more *attention* than the other, standard parts of the ad, as they are required to contain concise information about the product, service or brand in a very limited space of 70 characters. This information might also include special features about the advertised product, service or brand, such as shipping offers or discounts or even various characteristics of it that involve more specific and relevant knowledge of its application field. A reminder here is that the ad has to be as *relevant and informative* as possible towards the combination of both keyword and landing page for the promoted product in order to gain a good Quality Score, regardless its small length. Lastly, ad creatives need to have a *marketing gloss* that invokes the user's interest and urges him/her to click on the ad.

A usual practice of the advertisers is to exploit a feature of ad platforms called *keyword insertion* into ad-texts. This allows an advertiser to dynamically update the ad text with the keyword that is used to target the ad. A restriction with this approach is that it depends on a very specific template, changing only a minor part of the ad, while in many cases can result to an ad creative unreadable and not comprehensible. Thus, another important aspect that must be reserved in the ad-text creation process is that of *clarity*.

4.1.2 Pointwise Mutual Information Background

Point-wise mutual information (PMI) measures the strength of association between words. This is based on the fact that a simple but effective approach to collocation identification is to compare the *observed chance* of observing a combination of two words to the *expected chance*. This can be interpreted in various ways.

In [22, 68] PMI is defined as *"Document-based PMI"* :

$$dPmi(w_i, w_j) = \log_2 \frac{d(w_i, w_j)}{\frac{d(w_i)d(w_j)}{D}} \tag{4.1}$$

where D total number of documents in the corpus and the normalization factor in this case, $d(w_i, w_j)$ is the total number of documents in the corpus having at-least one span-constrained occurrence of the word pair (w_i, w_j), and $d(w_i)$, $d(w_j)$ are the total number of documents in the corpus containing at least one occurrence of w_i and w_j respectively.

In [20, 22] PMI is defined as *"Word-based PMI"*:

$$wPmi(w_i, w_j) = \log_2 \frac{f(w_i, w_j)}{\frac{f(w_i)f(w_j)}{W}} \tag{4.2}$$

where W is the total number of words in the corpus and the normalization factor in this case, $f(w_i, w_j)$ is the span-constrained (w_i, w_j) word pair frequency in the corpus, and $f(w_i)$, $f(w_j)$ are unigram frequencies of w_i and w_j respectively in the corpus.

4.1.3 Language Model Background

Several Natural Language Processing systems produce word sequences as their output, which have to be evaluated in terms of their likelihood i.e., how likely it is for them to occur in a sample text of their respective language. To ensure that the produced word sequences belong to a language, we first need to build a *language model (LM)*, [52, 67, 80] which is a probabilistic model of n-gram occurences and then validate the n-grams of the output sequence by checking if they reach a certain probabilistic threshold, according to the model. In speech recognition, it is traditional to use the term *language model* for a statistical model of word sequences.

Thus, in a simple formulation, LM is a mechanism for computing the probability of word sequences:

$$p(w_1, \dots, w_n) \tag{4.3}$$

LMs are an important component of speech recognition systems, a helpful mean to discriminate between similar sounding words, and reduce search costs. In statistical machine translation, a language model characterizes the target language, captures fluency and contributes in summarization.

The models of word sequences we will consider initially are probabilistic models; ways to assign probabilities to strings of words, whether for computing the probability of an entire sentence or for giving a probabilistic prediction of what the next word will be in a sequence. The simplest possible model of word sequences would simply let any word of the language follow any other word. In the probabilistic version of this theory, then, every word would have an equal probability of following every other word. If English had 100,000 words, the probability of any word would be $1/100.000$.

In a slightly more complex model of word sequences, any word could follow any other word, but the following word would appear with its normal frequency of occurrence. For example, the word *the* has a high relative frequency, it occurs 69,971 times in the Brown corpus of 1,000,000 words (i.e., 7% of the words in this particular corpus are *the*). By contrast the word *rabbit* occurs only 11 times in the Brown corpus.

We can use these relative frequencies to assign a probability distribution across following words. So if we have just seen the string *Anyhow*, we can use the probability .07 for *the* and .00001 for *rabbit* to guess the next word. But suppose we have just seen the following string:

Just then, the white

In this context *rabbit* seems like a more reasonable word to follow *white* than *the* does. This suggests that instead of just looking at the individual relative frequencies of words, we should look at the conditional probability of a word given the previous words. That is, the probability of seeing *rabbit* given that we just saw *white* (which we will represent as $P(rabbit|white)$) is higher than the probability of *rabbit* otherwise.

Given this intuition, this is why we presented above the probability of a complete string of words (which we can represent as $w_1, \dots, w_n$). These can have the role of a higher order n-gram or sentence (word sequence).

In an n-gram language model, we truncate the history to length $n-1$:

$$p(w_i|w_1,\dots,w_{i-1}) = p(w_i|w_{i-n+1},\dots,w_{i-1}) \tag{4.4}$$

where $p(w_1 w_2 \dots w_n)$ are the words that compose an n-gram. In practice, this calculation process can be exhaustive even for a computer. As a result, each probability is often approximated according to the **Markov Assumption**:

$$p(w_i|w_1 w_2 \dots w_{i-1}) \approx p(w_{i-k} \dots w_{i-1}) \tag{4.5}$$

which technically means that the occurrence probability can be calculated with the use of fewer k-grams. For example, we have the following cases:

- unigram model: $p(w_i)$
- bigram model: $p(w_i|w_{i-1})$
- trigram model: $p(w_i|w_{i-2}, w_{i-1})$

N-gram models can be trained by **counting** and **normalizing** (for probabilistic models, normalizing means dividing by some total count so that the resulting probabilities fall legally between 0 and 1). We take some training corpus, and from this corpus take the count of a particular bigram, and divide this count by the sum of all the bigrams that share the same first word.

We can simplify the calculation, since the sum of all bigram counts that start with a given word w_a must be equal to the unigram count for that word w_a.

Thus, we can estimate n-gram probabilities by counting relative frequency on a training corpus. The following is a maximum likelihood estimation which will be explained below.

$$\hat{p}(w_a) = \frac{c(w_a)}{N} \tag{4.6}$$

$$\hat{p}(w_b|w_a) = \frac{c(w_a, w_b)}{\sum_{w_b} c(w_a, w_b)} \approx \frac{c(w_a, w_b)}{c(w_a)} \tag{4.7}$$

where N is the total number of words in the training set and $c(\cdot)$ denotes count of the word or word sequence in the training data.

Equation 4.7 estimates the n-gram probability by dividing the observed frequency of a particular sequence by the observed frequency of a prefix. This ratio is called a **relative frequency**; the use of relative frequencies as a way to estimate probabilities is one example of the technique known as **Maximum Likelihood Estimation**, because the resulting parameter set is one in which the likelihood of the training set given the model is maximized.

The key-concept of the above language modeling methodology is that probabilities of an n-gram are calculated by the distinct probability of each word and therefore, when a word has zero occurrences $c(w_a) = 0$, n-grams containing it will get an undefined probability, which is problematic. This means that probabilities of n-grams that contain a low-probability word may be lowered as well, which does not really favour the language model, making it *sparse* [51]. Human language has "lopsided sparsity"; there is a fairly high probability of seeing an event that was not seen in the training corpus, even for large corpora. To overcome this problem, we often try to re-evaluate probabilities by using various *smoothing* techniques, which roughly fall into the following categories [45]:

Add-One or Laplace smoothing It is the simplest smoothing technique which normalizes probabilities by adding one pseudo-occurrence to each word count. C is the total number of word occurences in a total of $|V|$ number of different words and $c(w_a)$ the number of occurences of a single word, the new word occurence probability is calculated as follows:

$$p_{Laplace}(w_a) = \frac{c(w_a) + 1}{C + |V|} \tag{4.8}$$

Interpolation This is a weighted combination of target and lower-order distributions

$$p(w_i|w_{i-2}, w_{i-1}) = \lambda_3 f(w_i|w_{i-2}, w_{i-1}) + \lambda_2 f(w_i|w_{i-1}) + \lambda_1 f(w_i) + \lambda_0 \frac{1}{V} \quad (4.9)$$

where $f(w|\cdot)$ is a relative frequency estimate and $\sum_i \lambda_i = 1$. The weights are typically estimated on a held-out data set.

Backoff A technique that steals from the seen events and give to the unseen

$$p(w_i|w_{i-2}, w_{i-1}) = \begin{cases} f(w_3|w_1, w_2) & \text{if } c(w_1, w_2, w_3) \geq K_2 \\ discount(f(w_3|w_1, w_2)) & \text{if } K_1 \geq c(w_1, w_2, w_3) < K_2 \\ distribute(f(w_3|w_2)) & \text{if } c(w_1, w_2, w_3) \ < K_1 \end{cases} \quad (4.10)$$

Discounting can take different forms:

- absolute: subtract counts $(r - \delta)/N$
- linear: subtract a percentage $(1 - \alpha)r/N$

where r/N is the relative frequency estimate. Distributing spreads the stolen mass according to lower order distributions.

Chen and Goodman in their study [17] look extensively at different alternatives, testing with different amounts of training data and different corpora. The best results under a broad range of conditions are obtained using modified **Kneser-Ney smoothing**, thus for example for **trigrams** we have the following:

$$p_{KN}(w_i|w_{i-2}, w_{i-1}) = \frac{c(w_{i-2}, w_{i-1}, w_i) - D(c(w_{i-2}, w_{i-1}, w_i))}{c(w_{i-2}, w_{i-1})} + \gamma(w_{i-2}, w_{i-1}) p_{KN}(w_i|w_{i-1}) \quad (4.11)$$

where $c(w_{i-2}, w_{i-1}, w_i)$ are counts that actually represent the co-occurences of the unigrams, $\gamma(w_{i-2}, w_{i-1})$ is chosen such that the distributions sum to 1, and $D(c(\cdot))$ allows you to have smaller discounts for smaller counts.

Language modeling toolkits implement several options for smoothing; The SRILM toolkit [93] is one of the most commonly used LM toolkits [1].

Entropy and **perplexity** are the most common metrics used to evaluate n-gram systems. Corpus-based language models like n-grams are evaluated by separating the corpus into a training set and a test set, training the model on the

[1] http://www.speech.sri.com/projects/srilm/

training set, and evaluating on the test set. The entropy H, or more commonly the perplexity 2^H (more properly cross-entropy and cross-perplexity) of a test set are used to compare language models.

Entropy is a measure of information, and is invaluable in natural language processing, speech recognition, and computational linguistics. It can be used as a metric for how much information there is in a particular grammar, for how well a given grammar matches a given language, for how predictive a given n-gram grammar is about what the next word could be. Given two grammars and a corpus, we can use entropy to tell us which grammar better matches the corpus. We can also use entropy to compare how difficult two speech recognition tasks are, and also to measure how well a given probabilistic grammar matches human grammars.

Computing entropy requires that we establish a random variable X that ranges over whatever we are predicting (words, letters, parts of speech), and that has a particular probability function, call it $p(x)$.

The **cross entropy** is a useful concept when we do not know the actual probability distribution p that generated some data. It allows us to use some q, which is a model of p (i.e., an approximation to p). What makes the cross entropy useful is that the cross entropy $H(p,q)$ is an upper bound on the entropy $H(p)$. This means that we can use some simplified model q to help estimate the true entropy of a sequence of symbols drawn according to probability p. The more accurate q is, the closer the cross entropy will be to the true entropy. Thus, the difference between them is a measure of how accurate a model is. Between two models, the more accurate will be the one with the lower cross-entropy.

Considering the above, LMs are usually evaluated on their own in terms of *perplexity* [16]:

$$PP = 2^{\tilde{H}_r} \quad \text{where} \quad \tilde{H}_r = -\frac{1}{n}\log_2 p(w_1, \ldots, w_n) \tag{4.12}$$

where $\{w_1, \ldots, w_n\}$ is held out test corpus that provides the empirical distribution $q(\cdot)$ in the cross-entropy formula

$$\tilde{H} = -\sum_x q(x)\log p(x) \tag{4.13}$$

and $p(\cdot)$ is the LM estimated on a training set.

A lower entropy rate means that it is easier to predict the next symbol and hence easier to rule out alternatives when combined with other models, thus:

$$\text{small } \tilde{H}_r \rightarrow \text{ small } PP \tag{4.14}$$

When a distribution is uniform for a vocabulary of size V, then entropy is $\log_2 V$, and perplexity is V. So perplexity indicates an effective next-word vocabulary size, or branching factor. Minimizing $\tilde{H}_r$ is equivalent to maximizing log likelihood, and one commonly used model selection criterion (in general, not just for LMs) is maximum likelihood on held out data.

4.2 Related Work

In general, snippet generation for advertising purposes has been studied from another perspective, that of the auctioneer who is in need of a more efficient mechanism for matching keywords and ads, focused on contextual advertising tasks [73, 81]. The difference of contextual advertising [12] to that of sponsored search advertising is that it refers to the placement of commercial textual advertisements within the content of a generic webpage. However, the challenge from the auctioneer's point of view is similar, due to the responsibility of matching properly the ad to a given context (webpage or SERP).

Automatic summarization for matching ads. Anagnostopoulos et al. [5, 6] propose a method for just-in-time contextual advertising through web-page summarization, given the scenario that a web site features a spot for placing ads. The idea behind the method is the matching of existing ads to a landing page's summary, so that the overall process may happen on the fly while the web site that hosts the landing page loads. Summaries are constructed from various markup clues typically found in web pages i.e., HTML elements including meta-data keywords on one hand and external knowledge from the page URL with respect to a large taxonomy of advertising categories on the other. This matching can be achieved via the use of bag of words and classification features. The method suggested is a thorough example of web site summarization and its use for information extraction from product, service or brand descriptions. However, it does not provide much insight on the matter of ad creative generation, since it clearly proposes a page-ad matching routine alone. Another relevant task for a search engine is that of snippet generation for represent concisely a website to the organic results. Zhang et al. [109] propose a method for this task applying natural language processing techniques to extract and classify narrative paragraphs from the website and return key-sentences. This approach due to the formation of the goal results to a strict sentence extraction only for informative purposes.

Hui et al. [47] focus on selecting which ads will appear in the SERP by incorporating entity relationships as features in the decision mechanism from keyword statistics logs, thus their work is more relevant to that of proposing a quality ranking score. Choi et al. [19] explore the use of the landing page content in the ad selection mechanism. The authors use the ad context (including its title, creative, and bid phrases) to help identify regions of the landing page that should be used by the ad selection engine.

Handling short-text documents. Another interesting stream of work tackles with the increasing popularity of micro-blogging and user reviews posting services and studies classification and clustering techniques for short text documents (i.e., *snippets*) [82]. The major challenge in tasks that handle short text documents is to deal with the sparsity of the words in them and the possible lack of enough information and direct sources to distinguish properly the corresponding classes, clusters, or topics discussed in the snippets set. Yan et al. on their work at [104] deal with the topic modeling problem in terms of capturing topics from short texts. They explain the data sparsity problem on the effort of directly applying conventional topic models on short texts, thus instead they propose a biterm topic model which relies on the co-occurences of any two distinct words, namely bigrams and the adoption of Gibbs sampling to perform approximate inference. Louis et al. [61] propose a methodology for business-related tweets summarization based on concepts. For a specific brand or company name they retrieve relevant tweets and their goal is to summarize information and events relevant to each mentioned company. First, they retrieve different kinds of business-related concepts from free text and then they cluster the tweets according to each learned concept. They rank the mentioned company tweets inside each cluster based on information and sentiment analysis. This follows the logic of aspect-based summarization of product reviews. Ma et al. [63] try to summarize comments and articles from news sites for each referenced topic, hence they deal with topic modeling first. In order to select the most representative comments, they utilize two different comment ranking schemes than the existing mechanisms due to the "rich-get-richer" problematic effect of awarding the most popular comments.

Hu et al. in [46] propose a framework for improving the performance of short text clustering by exploiting the internal semantics from the original text and external concepts from world knowledge. Song et al. in [90] try to conceptualize (instead of topic modeling they try to classify the snippets according to concepts like country or region) short texts using a Bayesian inference mechanism. They utilize Probase[2] as a knowledge base for the concepts.

Language models for product categories. Gopalakrishnan et al. in [43] propose an unsupervised algorithm for matching product titles from different data feeds that refer to the same underlying product entity which is a key problem in online shopping. They leverage a search engine to enrich product titles by adding important missing features that occur frequently in search results and then they compute importance scores for these features based on their identification power. Shen et al. in [89] leverage a graph algorithm for classify product items into a large taxonomy of categories instead of relying on a human-defined hierarchy to organize the corresponding structure. In this way they discover automatically groups of highly similar classes. The notion of using a per-topic language model is introduced in the work of Lin et al. [59]. They approach the problem of tracking topics in a continuous stream of short language texts with two combined language models: a foreground model to capture recency and a background model to combat data sparsity experiment with different smoothing techniques for building the language models. A topic-focused summarization methodology is proposed by Sood et al. in [91]. In a first phase they process web documents through a topic modeling task to find the primary topic of discussion in chats. In a second phase they use a co-occurence Hyperspace Analogue to Language (HAL) Language Model to score their candidate sentences.

Regarding the automated ad-text generation process for online advertising campaigns, this issue remains still an open problem as mentioned in [13, 34]. Bartz

[2] http://research.microsoft.com/en-us/projects/probase/

et al. [8] approach the issue by using strict ad templates with slots reserved for bidding terms that need to be carefully reformatted to suit the given template. Fujita et al. [33] attempt to generate shop-specific listing ads (in japanese) by reusing previously written text ads for them. A similar task can be found in the work of Ganesan et al. [36] where the authors present an unsupervised approach to generate ultra-concise summaries of opinions. They formulate the problem of generating such a micropinion summary as an optimization problem, where they seek a set of phrases that represent key opinions in text and propose some heuristic algorithms.

Recently, AdWords introduced a feature called *Dynamic Search Ads* [3]. This feature modifies the ads that have been set in a campaign based on the different kinds of searches a user makes. However, it follows an approach that enhances only some parts of the ad (similar to the keyword insertion functionality, e.g., it can modify a certain defined term for changes) that it is different from the aim of our task. In addition, it needs a corpus of many landing pages inside a website thus it could not be used by websites with less than some hundred landing pages.

4.3 Phrase Extraction

In an initial stage, the goal is to gather information about the promoted product. Our only source for this kind of information is the landing page of the product, from which we need to obtain the most important parts that must be conveyed into the generated corresponding ad-text. In this task, we assume that we have an already preprocessed landing page and the corresponding segments that we need i.e., description and possible contained reviews and comments.

4.3.1 Representativeness Property

Our work is based on the *representativeness* definition of Ganesan et al. [36]. In general, some text in the description field as well as in the contained reviews are often redundant and may contain contradicting viewpoints. Hence, generating a few highly representative phrases is a challenge. Since we are mainly

[3] https://support.google.com/adwords/answer/2471185?hl=en

interested in summarizing the major points in text, a representative summary could defined as the following:

> *"A representative summary would be one that can accurately bring to surface the most common praise or critical information."*

For example, assuming we have 10 sentences in the input document that talks about "beautiful nexus 7 screen" and one about "new nexus 7 screen", by our definition, the former would be the representative summary phrase.

In order to extract phrases in the context of ad-text generation we followed initially an unsupervised method. The method relies on the construction of a unigram model, while extending it with higher order n-grams until there is no further production of new n-grams available. At every step of forming newer n-grams, non-promising candidates are pruned by calculating the *representativeness* score. Words used in phrases should not only be frequent in text, but also strongly associated. Thus, we need to employ a good measure for extracting collocations, but at the same time maintain a decent level of "topical coherence".

Remark 4.1. We consider the content of a landing page (our corpus) as a set of sentences. Bigrams can be found in the context of these sentences.

- Each sentence and generated candidate phrase can be composed by n unigrams $w_1, ..., w_n$
- In our work, we consider only those word-pair collocations where inter-word distance (i.e., span or context-window) between w_i and w_j is exactly 1. For us, a bigram (w_i, w_j) is a sequence of two adjacent unigrams which are collocating exactly the one beside the other inside the limitation of a sentence
- Order matters: w_j is following w_i
- $f(w_i, w_j)$ is the window-constrained (w_i, w_j) word pair frequency in the corpus (in our task this matches the definition of bigram frequencies)
- $c(w_i, w_j)$ is another smaller distance constraint word pair frequency (in our experiments we set this distance to 1, but can be modified according to the task)

- $f(w_i)$, $f(w_j)$: unigram frequencies of w_i and w_j respectively in the landing page
- The normalization factors here are U, which is the total number of unique unigrams and B which is the total number of unique bigram, thus we consider the following: $P(w_i, w_j) = f(w_i, w_j)/B$, $P(w_i) = f(w_i)/U$, and $P(w_j) = f(w_j)/U$

Definition 4.2 (Weighted word-based PMI)**.** We define a modification of the word-based PMI notion, where the "weight assignment" in the numerator rewards well associated words with high co-occurence.

$$wwPmi(w_i, w_j) = \log_2 \frac{P(w_i, w_j) \cdot c(w_i, w_j)}{P(w_i) \cdot P(w_j)} \tag{4.15}$$

The co-occurrence frequency, $c(w_i, w_j)$ which is not part of the original PMI formula is integrated into the PMI scoring to reward frequently occurring words from the original text. The problem with the original PMI scoring is that it yields in high scores for low frequency words. By adding an extra weight factor into the PMI scoring, we ensure that low frequency words do not dominate and moderately associated words with high co-occurrences have relatively high scores.

Then, the *representativeness* score of each phrase is the average $wwPmi(w_i, w_j)$ of every two unigrams that co-occur next to each other in the phrase and N is the number of such pairs.

$$Representativeness(w_1, ..., w_n) = \frac{1}{N} \sum_{i,j} wwPmi(w_i, w_j) \tag{4.16}$$

4.3.2 N-grams Formulation

At a high level, in the Phrase Extraction (PE) phase, we start with a set of high frequency unigrams U from the original text. We then gradually merge them to generate higher order n-grams as long as their representativeness remains reasonably high. This process of generating candidates stops when an attempt to grow an existing candidate leads to phrases that are low in representativeness (i.e., do not satisfy the threshold σ_{rep}[4]).

Using bigrams B as seed candidates, we concatenate each candidate that has representativeness score higher than σ_{rep} with another n-gram that shares an overlapping word, meaning that the ending word in $ngram_1$ should overlap with the starting word in $ngram_2$. In addition, $ngram_1$ should not be a "mirror"[5] of $ngram_2$. Furthermore, we use jaccard similarity to avoid redundancies in our candidates list. Specifically, before adding a candidate phrase X to the list, we check whether there is another phrase in the list that is similar to X. Candidates should have a jaccard similarity lower than a threshold $\sigma_{jaccard}$[6]. If a similar phrase in our candidates list is found then we keep the candidate that has a higher representativeness score. We repeat this recursive process until no possible expansion of our candidates list.

4.4 Advertising Language Generation

In the phase of the Advertising Language Generation (ALG), we make the proper transformations in order to follow the aforementioned properties that an ad-text must have.

4.4.1 N-grams Transformation

From the n-grams obtained at the Phrase Extraction (PE) stage, we need to choose only a few best to be included in the ad-text, due to length and significance restrictions. To do so, we calculate the mean score of the entire set and then choose the n-grams with score higher or equal to the mean. Those with length greater than 70 characters are also removed from the final set. This limit is not arbitrary, as most search engines allow text ads with this specific character limitation.

The method continues with trimming the n-gram set by performing heuristic grammatical corrections. We eliminate n-grams of low readability: n-grams that

[4]After parameter tuning, in our experiments we set $\sigma_{rep} = 3$
[5]Example: "new phone" is a mirror to "phone new"
[6]After parameter tuning, in our experiments we set $\sigma_{jaccard} = 0.7$

contain sequences of 5 or more nouns in a row and might have been erroneously extracted from the landing page [7]. If an n-gram contained an adjective at the end, we relocated it at the beginning.

Next, we build *permutations* of the best n-grams in order to add them to the final representation, assuring the character limitation mentioned before.

At this point, it is essential to picture the final representation of the snippet as a collection of empty slots. The information that needs to be conveyed includes: the product name, a feature sequence about the product and its price (optionally). Our method proposes two simple sentence templates which provide the necessary slots:

<product name> with <feature set> <price> (a)

<feature set> <price> (b)

To generate possible candidate ad snippets, we first fill in the <feature set> slot with all n-grams sequences generated above. We add the extracted product name, if it is given, to the corresponding slot of (a), once again with respect to the size of the generated sentence. In addition, we can concat the product price - if available, provided that there is space left for this addition.

This stage often results in a large number of candidates, as a consequence of the number of feature sets or the length of the other features. We consider the problem of pruning the output candidate set, in pursuance of retrieving a small amount of high-quality candidates. The idea here is to rank candidates using two proposed scoring functions: the *Information Scoring* function and the *Readability Scoring* function.

4.4.2 Information Property

It is not uncommon that some generated candidates in the result set do not include the most significant product features, i.e., highest-score n-grams which

[7] We used the Stanford Part-of-speech Tagger http://nlp.stanford.edu/software/tagger.shtml

were extracted from the landing page. Moreover, some others do not utilize the 70-character space efficiently, resulting in extremely small and inadequate ads. Therefore we implement a scoring function that measures the overall Information Gain from the snippet and penalizes the candidates with little utilization of space. For the second template (b), given a candidate c with s_i being the representativeness scores of the n-grams included in it, n the number of features and $l(c)$ the length of the candidate, the value of the Information Scoring Function I_i is:

$$I(c) = \frac{1}{n + \frac{70}{l(c)}} \sum_{i=1}^{n} s_i \tag{4.17}$$

For the first template (a), the appearance of the product name is considered to be very significant for the general completeness of an advertising text. The idea here is to promote candidates that feature the product name by taking it into account as a multiplying factor the product feature with score equal to the maximum of the set, $\max s_i$. To meet this standard, we adapt the Information Scoring Function as follows:

$$I(c) = \frac{1}{n + \frac{70}{l(c)}} \max_{1<i<n} s_i + \sum_{i=0}^{n} s_i \tag{4.18}$$

4.4.3 Readability Property

The Readability Scoring Function awards candidates that fulfil two important conditions: they are readable according to advertising standards and grammatically correct. This measure fully relies on the application of an "advertising language model", which is language model trained on an ad-text dataset.

Constructing the ad dataset

Our initial approach was to build an ad dataset of 47,984 unique ads obtained from major search engines. We queried search engines using as phrases the titles of the nodes from Google Products Taxonomy[8], which is a taxonomy of 21 major product categories and 5487 sub-categories. We expanded these phrases by selecting similar categories and keywords from the Google Keyword Suggestion Tool and used them both as broad as well as phrase match queries (i.e., using quotes or not). Each query returned a number of ads, from which we

[8] https://support.google.com/merchants/answer/1705911

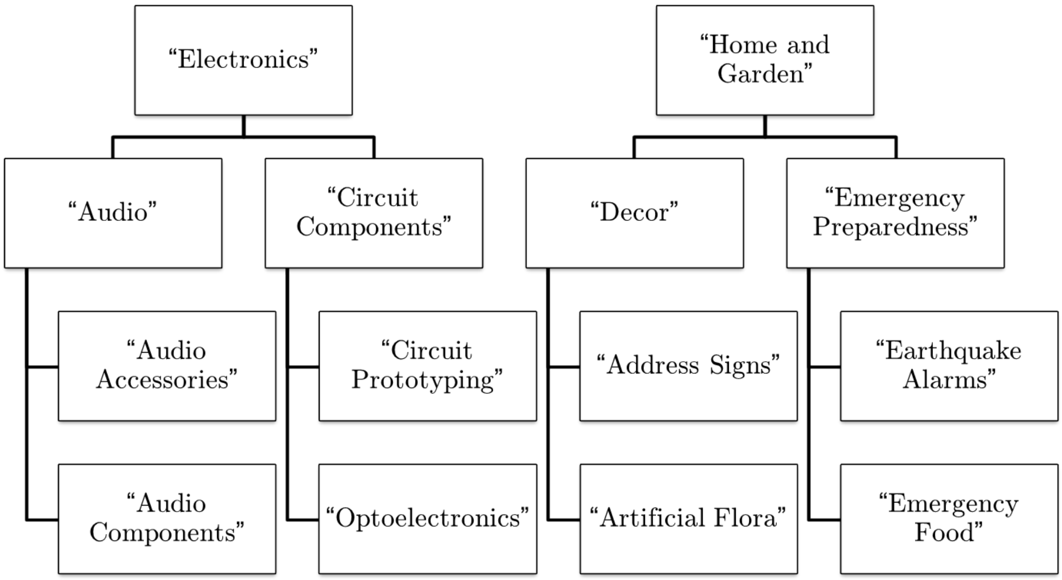


FIGURE 4.1: Taxonomy Terms Snapshot

Class	Snippet
Apparel & Accessories	Sweaters & Knits Cardigans - Spring / Summer 2013. Shop the Season's Hottest Trends. Over 500 Brands and Free Shipping!
Business & Industrial	Advertising & Marketing. What is Advertising & Marketing? Understand more about it here!
Cameras & Optics	Cheap Camera Lenses. Best Buy Camera Lenses Discount Price Special Offers
Electronics	High Voltage Diodes. Manufacturer of High Voltage Diodes rectifiers and assemblies
Home & Garden	Industrial Water Heaters. #1 brand in water heaters for the concrete industry. Custom Options.

TABLE 4.1: Example of retrieved class-snippet pairs

kept all integral parts, along with the path of the query that triggered them in the taxonomy and the base class. In Figure 4.1 we present a snapshot from the taxonomy and in Table 4.1 an example snapshot of the gathered class-snippet pairs. We fed SRILM Toolkit with the snippets as input corpus and built a language model based on trigrams and on the Kneser-Ney discounting method.

Note that due to the relatively small size of the ad dataset, it was fairly natural for any test data used afterwards to include several unknown -to the model- word tokens. Therefore, we indicate with a parameter that our model should treat any unknown word as a regular word. Also, the choice of the discounting method was based on the same observation. The Kneser-Ney discounting method did not seem to "punish" unknown word tokens as much as other discounting methods, also assigning better probabilities to them. For each candidate we keep its **logarithmic probability** as the value of the Readability Scoring Function, which is an indication of the likelihood that a given candidate will occur, according to the language model.

4.4.4 Candidate Pruning and Normalization

It is important to notice that each scoring function produces scores of different magnitude. To combine these scores we used the Min-Max Normalization Technique to bring all scores in the [0,1] range and find those candidates with a better mean than others. Depending on the number of candidates that we want to keep, we can prune after a certain threshold. The normalized Information and Readability scores have equal contribution in the overall score of a candidate, thus we set $\alpha = 0.5$ in the following equation.

$$OverallScore(c) = \alpha \cdot I_{norm}(c) + (1 - \alpha) \cdot R_{norm}(c) \tag{4.19}$$

4.4.5 Sentiment Analysis

Sentiment analysis in our method aims to determine the contextual polarity of a phrase. In this task, we used a snapshot from the Amazon Sentiment Dataset[9]. The dataset consists of Amazon reviews with a 5-star rating. Neutral reviews with 3 stars were omitted during the construction of Amazon dataset by its authors. Since this dataset does not contain any neutral reviews, we could easily form two categories: positive reviews and negative reviews, tagged **'pos'** or **'neg'** respectively. Each line in the positive and negative set corresponds to a single snippet, containing roughly one single sentence.

[9]Large-Scale Datasets for Sentiment Analysis http://mst.cs.drexel.edu/datasets/SentimentDatasets/SentimentDatasets

Measure	All words	top 1000 words	top 10000 words	top 15000 words
Precision (pos)	0.913	0.938	0.915	0.913
Recall (pos)	0.756	0.566	0.722	0.736
Precision (neg)	0.791	0.689	0.770	0.779
Recall (neg)	0.927	0.963	0.933	0.929
Accuracy	**0.842**	0.763	0.828	0.832

TABLE 4.2: Sentiment Analysis Feature Selection

Separating this dataset into train set and test set, we train a simple Naive Bayes classifier[10] in order to have a first look at the support of the sentiment analysis procedure. We trained on 261,346 instances, and tested on 87,116 of the dataset reviews.

For this classification task we experiment in feature selection. Initially, all words in each review were our features. In the next experiments we select the top k informative words as features, by measuring the Information Gain of each word. Using all words as features provided better accuracy in the test set than choosing the top-k informative words, as Table 4.2 demonstrates, thus we proceeded our classification model with all words.

Inside the landing page -if for example the website has a dynamic content and in a specific form lets users leave a free-text review- might exist also negative comments. The goal of the sentiment analysis component is to leverage the trained model and filter out possible negative snippets.

4.5 Comparative Evaluation

We wanted to investigate if the resulted snippets could be actually used as convincing ad-texts of a campaign, thus we defined appropriate criteria for this kind of purpose. In this experiment we obtained judgements from evaluators in order to compare the performance in each property for different methods.

[10]We used Natural Language Toolkit implementation `http://nltk.googlecode.com/svn/trunk/doc/api/nltk.classify.naivebayes.NaiveBayesClassifier-class.html`

4.5.1 Experimental setup

In order to examine the potential of the proposed method, we experimented with variations of our principal method. In the following paragraphs we present our results as well as a comparative evaluation with baseline approaches. We experimented on 100 product landing pages from two major online catalog aggregators, 50 from eBay[11] and 50 from BestBuy[12]. The pages were equally distributed among a subset of available product categories. For each landing page we applied 7 methods and we kept the top-3 text ads (i.e., ad snippets) resulting to 2100 snippets for evaluation.

The evaluation was held following a blind testing protocol by two groups of 6 human evaluators each, a total of 12 volunteer researchers. In each group was given triplets of the form *{landing page, product name, ad snippet}*. Group A had to evaluate all of the 2100 instances, meaning each person had to evaluate 350 snippets. Group B had to do the same. In this way we could measure the **interjudge agreement** between the answers of these two groups.

The evaluators were asked to answer if the ad snippet met a criterion or not and assign 1 or 0 respectively. The criteria were the following:

Attractiveness : Is the snippet attractive in order to prompt clicking on the ad?

Clarity : Combination of Grammaticality and Readability.
Is the snippet structured in a comprehensible form?

Relevance : Is the snippet representative for the corresponding landing page of the promoted product?

As an overall score we calculate the **Harmonic Mean**, as we need an average of rates for the previous criteria.

Baselines

To assess how well our main approaches perform against simpler methods such as well tuned baselines with some heuristic checks, we use the following configurations of baselines. A note here is that we do not begin with stopword removal because we want to investigate if a more simplistic extraction approach

[11]http://www.ebay.com/
[12]http://www.bestbuy.com/

could select a representative sentence from the landing page as ad-text candidate.

- PE: In this method we only utilize the PE phase of our system. The extracted n-grams serve as ads. There is just one check regarding the overlap between different n-grams. In this premise, lower order n-grams that are included in higher order n-grams are removed from the candidate list.
- $PE + HG$: We expand PE with a heuristic grammar check using Penn Treebank to eliminate candidates that a. contain conjuction, determiner, or comparative adverb in the last position of the sentence b. the tree confidence score of the Stanford lexicalized parser does not surpass a certain threshold ($\tau = -87$).
- $PE + SA$: Combination of PE along with sentiment analysis, by removing any retrieved phrases that are classified as negative.
- $PE + HG + SA$: Combination of PE with HG and SA .
- $PE + CP$: After PE phase, proceed with the original form of the returned n-grams and follow a very basic candidate pruning by evaluating them with Information and Readability Scoring Functions

Principal Method Variations

Here, we introduce our integrated method. For every webpage in our dataset, we begin with stopword removal before proceeding to the PE phase. Additionally, we continue with the ALG phase ($PE + ALG$). Another variation is to extend $PE + ALG$ with the addition of the SA task, hence we remove any retrieved negative n-grams from the PE phase before we proceed to the ALG phase ($PE + ALG + SA$).

4.5.2 Human Evaluation Results

It is interesting as we mentioned earlier to measure the **interjudge agreement** i.e., how much agreement between judges there is on assigning scores for the same set. In Table 4.3 we present $P(D)$ (observed proportion of the times the judges agreed), $P(E)$ (proportion of the times they would be expected to agree

Criterion	$P(D)$	$P(E)$	kappa statistic
Attractiveness	0.891	0.553	0.756
Clarity	0.912	0.561	0.800
Relevance	0.944	0.541	0.877

TABLE 4.3: Interjudge Agreement

Method	Attractiveness	Clarity	Readability	Harmonic Mean
PE	0.387	0.677	0.660	0.538
PE+HG	0.253	0.693	0.517	0.410
PE+SA	0.433	0.697	0.680	0.575
PE+HG+SA	0.293	0.647	0.550	0.443
PE+CP	0.257	0.617	0.423	0.381
PE+ALG	**0.527**	**0.854**	**0.943**	**0.726**
PE+ALG+SA	**0.593**	**0.850**	**0.937**	**0.763**

TABLE 4.4: Criteria Rates Per Method

by chance), and *kappa statistic* [15, 23] (correction of a simple agreement rate for the rate of chance agreement) which is calculated as follows:

$$kappa = \frac{P(A) - P(E)}{1 - P(E)} \tag{4.20}$$

The $kappa$ value will be 1 if two judges always agree, 0 if they agree only at the rate given by chance, and negative if they are worse than random. As a rule of thumb, a $kappa$ value above 0.8 is taken as good agreement, a kappa value between 0.67 and 0.8 is taken as fair agreement, and agreement below 0.67 is seen as data providing a dubious basis for an evaluation. We observe in the results very high $kappa$ values. This could be an outcome of the fact that the evaluators had only binary choices, thus more strict decisions and in this way more similar behaviour on their judgements.

In Table 4.4 we present the criteria scores for each method and we observe that our principal method variations outperform the baselines. Some variations on baselines, which in first glance seemed that they would achieve better performance, actually went worse as they were eliminating candidates with very simple errors that seem not to influence significantly the evaluators.

We performed the Wilcoxon rank-sum test in order to investigate whether the improvement between the most efficient methods and the best performing baselines is statistically significant. As a result, the differences in each score between

Method	Product Name	Snippet
PE	Canon PIXMA iP100	*Auto image fix function automatically adjusts image*
PE+HG	Dell XPS Desktop	*Microsoft windows 8 64-bit operating system preinstalled*
PE+SA	Samsung Galaxy S	*Cell phone free shipping no contract required $25 free extras*
PE+HG+SA	Virgin Mobile - LG Optimus	*Bluetooth compatibility for wireless communication*
PE+CP	Dell Ultrabook	*Corning gorilla glass ensures durability*
PE+ALG	VIZIO ESeries HDTV	***Vizio eseries with effective refresh rate, low price guarantee***
PE+ALG+SA	Fujifilm Finepix JX580	***Artistically enliven photos, instantaneously increases shutter speed***

TABLE 4.5: Examples of generated promotional text from all methods with top scores

- PE and $PE + ALG$
- PE and $PE + ALG + SA$
- $PE + SA$ and $PE + ALG$
- $PE + SA$ and $PE + ALG + SA$

were actually statistically significant ($p < 0.05$). Adding SA does not affect the computational cost while it improves each time the overall score, thus it provides an interesting solution for more prompting ads. In Table 4.5 we present examples of generated promotional text that met all the criteria.

4.6 Language Models for Product Categories

In this section we present an extension of our initial approach. We aim at a more focused generation of ad candidates. On this premise, we propose a sequence of tasks for leveraging the specific category that a promoted product may belong. In order to make use of an appropriate corpus for this goal, we retrieved 34,527 unique ads from a major search engine and 55,655 short descriptions from a large electronic commerce website, making a total of 90,182 unique snippets. The tested search queries were all the nodes of the Google Products Taxonomy. For the e-commerce site source we kept only the first two sentences of the description in order to keep the snippet short and collect only

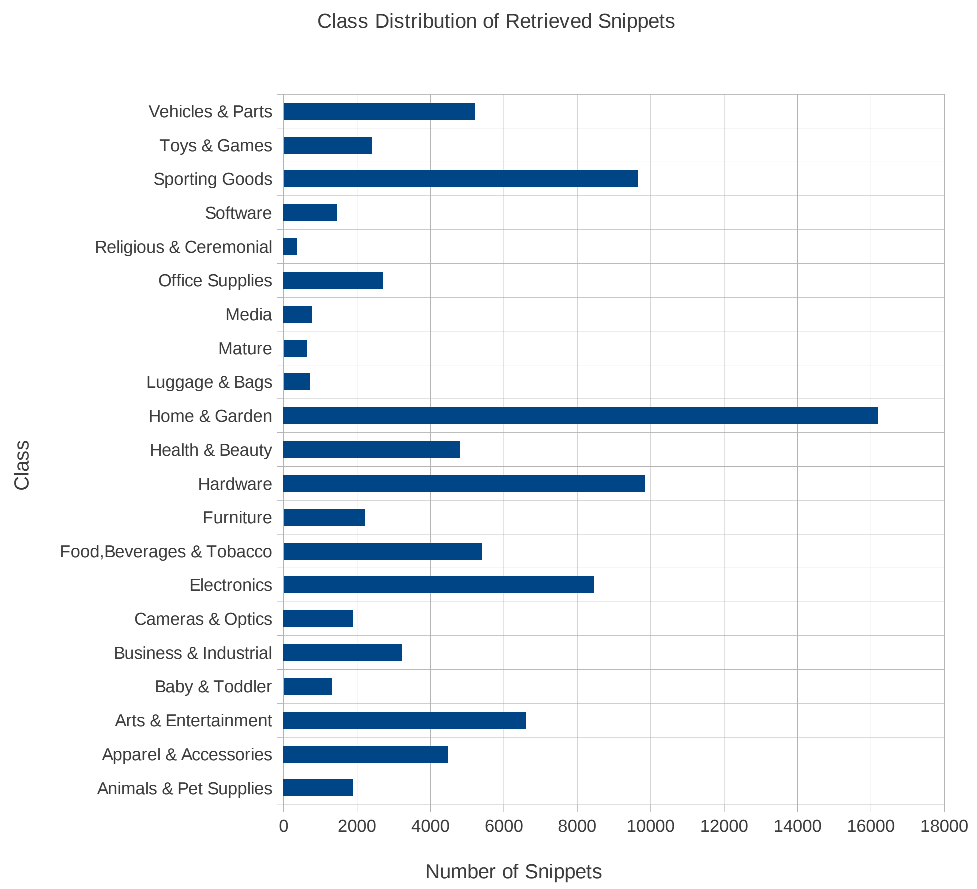


FIGURE 4.2: Class Distribution

the most important information and linguistic focus of the promotional text. In this way we achieve two objectives. We can construct distinct language models per class in order to have a more focused language for our generated snippets. These language models help us evaluate better the phrases in terms of their marketing appeal as well as their specific product category. In addition, we obtained ground-truth labeling class-snippet pairs which help us evaluate in our following tasks the accuracy of a short text multiclass classification. These language models are based as previously on trigrams and on the Kneser-Ney discounting method.

In Figure 4.2 we present the class distribution of our obtained dataset.

Training Instances	Testing Instances	Accuracy
80,182	10,000	75.94%
75,182	15,000	75.6%
70,182	20,000	75.34%
65,182	25,000	74.52%
60,182	30,000	74.72%

TABLE 4.6: Short Text Multiclass Classification Accuracy

4.6.1 Short Text Multiclass Classification Task

In this task we aim at classifying the given landing page to one of the 21 aforementioned classes in order to understand which language model we are going to exploit. The representation of the landing page as a snippet is created by keeping the 5 best n-grams of the Phrase Extraction phase. We wanted to use a method for multiclass text classification. The problem here is that we cannot refer to typical document classification methods as we face the challenge of the short text representation. Thus, we need a modified version of a multiclass classifier. For the purposes of our experiments we decided to use the LibShortText open source tool for short-text classification and analysis [44] [13] This multiclass classification engine is based on LIBLINEAR to support the one-versus-rest approach introduced in Bottou et al. [11] and the method by Crammer and Singer [21]. To test if the selected algorithm is efficient for our task we leverage the retrieved labeled snippets. The testing set each time is selected with stratified sampling. In 4.6 and 4.3 we present the training and testing statistic results. In general, for this kind of task an average of about 75 % accuracy is very good considering that we categorize snippets into 21 classes and not performing just a two-class classification. Similar findings have been mentioned in the work of Lee et al. [55] where they were performing multiclass topic classification with 18 categories for tweets.

[13]It can handle the classification of, for example, titles, questions, sentences, and short messages. It is more efficient than general text-mining packages. On a typical computer, processing and training 10 million short texts takes only around half an hour. The fast training and testing is built upon the linear classifier liblinear.

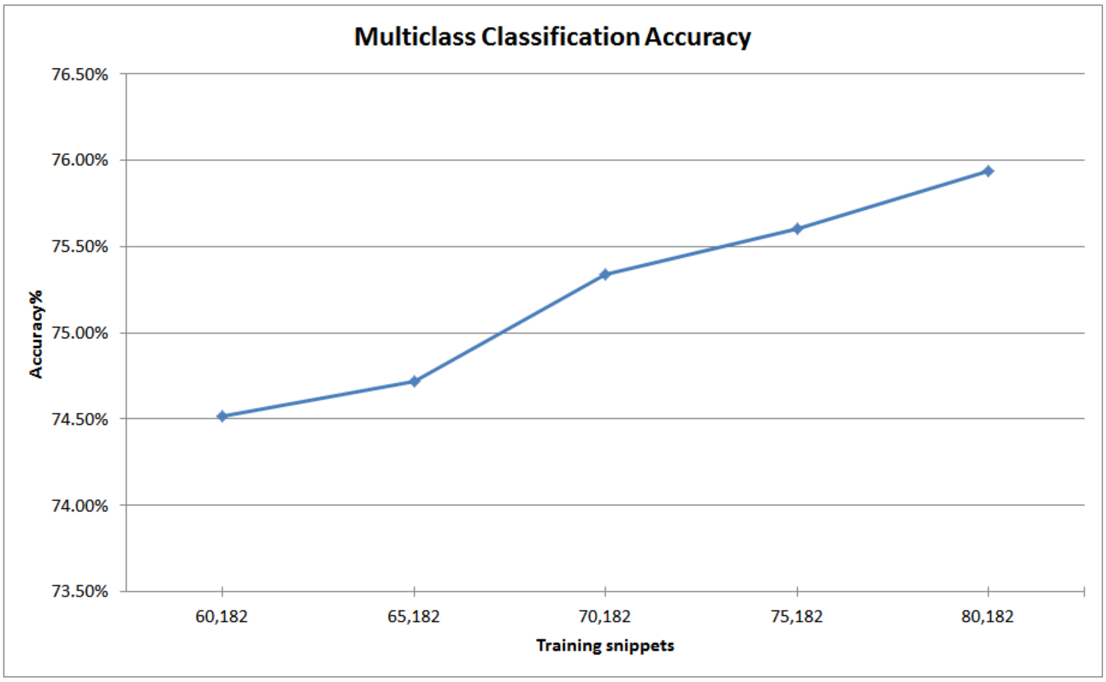


FIGURE 4.3: Multiclass Classification Accuracy

4.6.2 Real-world Campaign Experiments

In this section, we employ a setting of a real-world online advertising campaign using the Google AdWords Platform in order to evaluate in practice the resulted ad-texts. Because we want to isolate the bidding strategy from our problem we just use the exact same campaign, adgroup and keywords for the evaluated ads. We have chosen as the preferred ad delivery setting the evenly ad rotation option to give a fair chance to the ads to be displayed, following an A/B testing-like strategy. The campaign promotes a car rental website. We ran a campaign during October 11-14 and we set a daily budget of 25 euros. The challenge is to achieve a performance close to the human placed description.

We tested the following processes as ad-text generation competitors:

- *Human Description (HD)*, which is a human language text created by an advertiser.
- *Central Method (CeM)*, which is the main process that we have proposed in $PE + ALG + SA$.

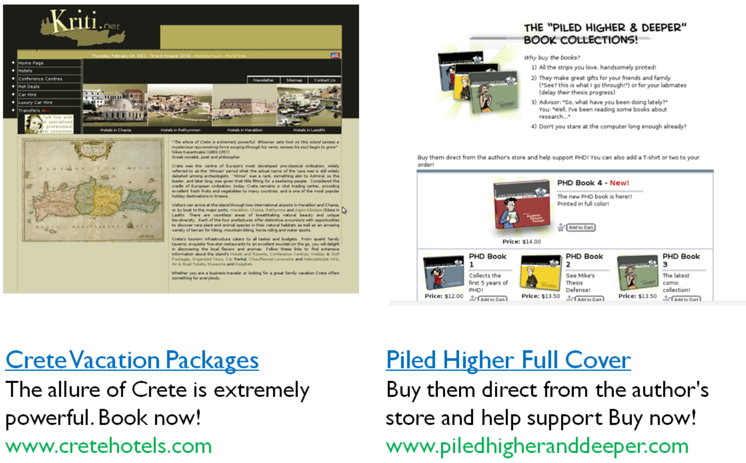


FIGURE 4.4: Heuristic baseline ad creative examples

- *Extended Method (ExM)*, which is the extension using a product category language model and short text classification. *ExM* in the classification task assigns the given landing page in the *Vehicles & Parts* class.
- *Baseline*: In addition, we have constructed a heuristic baseline text summarizer as a benchmark.

Heuristic Baseline Approach

We will describe here in short the baseline approach. The first step was to extract all the text from the HTML document of the given landing page. Then, we used summarization to keep the most important meaning for the description of our advertising page. For this purpose the input was the text from the page to the Classifier4J [14] which uses internally a Bayesian text classifier in order to select the most important sentences for the final summary. We kept the constraints and limitations that are given from Google AdWords platform (number of characters in the ad lines) and we added at the end of the second description line a call-to-action phrase such as: "Buy now!". In Algorithm 1 we present the process and in Figure 4.4 we present a sample of the baseline outcome.

In Table 4.7 we present for each competing method the generated ad-texts for the car rental partner. For each method we have selected a representative candidate.

[14] http://classifier4j.sourceforge.net/

Algorithm 1: Heuristic baseline ad-text creation

Input: Landing Page HTLM Document d, t the target of the advertisement
Output: An ad-text
Let λ be the length of a sentence in characters
Let ϕ be the set of the action phrases
$limit_1 = 25, limit_2 = 35$
▷ Choose a proper action phrase $p_{action} \in \phi$ with respect to t
$p_{action} \mapsto t$
$bidPhrase \leftarrow keywordGenModule(url_{destination})$
▷ Retrieve the first phrase of the title until the first punctuation
$title =< p_1, p_2, \ldots, p_n >$
if $\lambda_{p_1} < limit_1$ **then**
    $head \leftarrow p_1 \cap bidPhrase$
else
    $head \leftarrow bidPhrase$
$head \leftarrow capitalizeFirstLetterOfGrams(head)$
▷ Summarise d in 1 sentence using Bayesian classifier
$d_{summary} \leftarrow summariser(d, 1)$
$d_{summary} =< s_1, s_2, \ldots, s_N >$
while $\lambda_{\bigcap_{i \in N} s_i} \leq limit_2$ **do**
    $dl_1 \leftarrow \bigcap_{i \in N} s_i$
end
while $\lambda_{(\bigcap_{k \in N} s_k) \cap p_{action}} \leq limit_2$ **do**
    $dl_2 \leftarrow \bigcap_{k \in N} s_k$
end
if s_{final} *is a stopword* **then**
    remove s_{final} from dl_2
$dl_2 \leftarrow (dl_2 \cap p_{action})$
$url_{display} \leftarrow "www." \cap p_1 \cap ".com"$
$adText \leftarrow (head \cap dl_1 \cap dl_2 \cap url_{display})$
return *adText*

Method	Snippet
Baseline	Athens is the capital and largest city of Greece and Book now!
HD	Very cheap prices and reliable services for rental worldwide !
CeM	Travelers Guide Information, central Aegean Sea.
ExM	Fine white sandy beaches, popular modern day tourist destinations.

TABLE 4.7: Generated Ad-texts

Method	CTR	Clicks	Impressions	%Served	Avg. CPC	Cost	Avg. Pos.
Baseline	0.20%	2	1019	5.89 %	1.46	2.91	5
HD	0.72%	33	4606	26.63%	0.90	29.57	5.6
CeM	0.57%	30	5223	30.2%	1.15	34.52	5.1
ExM	**0.59%**	36	6054	35%	**0.96**	**34.41**	5.3

TABLE 4.8: Performance Comparison

In Table 4.8 we demonstrate the performance of the competing methods. The reader is reminded that CTR represents how often people click the ad after it is shown to them. This is the first criterion for the performance in this experiment as we wanted to see if the automated text was approaching the human performance. We notice that the *ExM* outperforms the baseline summarizer and the *CeM* method, converging to the human language description performance. The second criterion is the Average cost-per-click (Avg. CPC). This is the average amount that the advertiser has been charged for a click on the ad. This amount is the total cost of all clicks divided by the total number of clicks received. We notice here that the auctioneer charges with smaller cost our *ExM* method (actually very near to the cost of the *HD*), a fact that means that this ad has achieved a good *Quality Score*. This metric is based both on the CTR performance as well as the relevance of the ad with the keywords of the AdGroup and the landing page.

Finally, we can see the percentage of the ad delivery quota (%Served). We infer from this rotation, that the optimization mechanism of the auctioneer's platform has selected as a good performing ad-text, the *ExM* generated. On average, all of the ad-texts were placed at the 5th position of the auction. This is a very satisfactory result given the competitiveness of the chosen business field (car rental service).

Chapter 5

Budget Optimization

In this chapter, we tackle the task of *Campaign Optimization*, thus we study the corresponding problem of *Budget Optimization* [58].

5.1 Introduction

As we discussed earlier, search engines commonly use Pay Per Click (PPC) auctions to sell their available inventory of ad positions for any search query. In these auctions, advertisers select keywords of interest, create brief text ads for the keywords and submit a bid for each keyword which indicates their willingness to pay for every click. For example, a car rental company may submit the following set of keyword - bid pairs {*(car rental Greece, $2), (cheap car rental Crete, $5), (rent car Crete, $3),....*}. Large advertisers typically bid on hundreds of thousands of keywords at any instant. When a user types a query, the search engine identifies all advertisers bidding on that (or a closely related) keyword and displays their ads in an ordered list. The search engine uses the advertisers' bids along with measures of ad relevance to rank order the submitted ads. Whenever a consumer clicks on an ad in a given position, the search engine charges the corresponding advertiser a cost per click (CPC) which is the minimum bid needed to secure that position. The auctions are continuous sealed bid auctions. That is, advertisers can change their bids at any time and cannot observe the bids of their competitors. Typically advertisers are only given summary reports with details such as the total number of impressions, clicks and conversions, average rank and average CPC for each keyword on a given day. Several of these auctions are very competitive. For example, it is not uncommon to have 100 or more advertisers bidding for the same keyword. The average CPC on search engines has been continually rising over the last couple of years and search advertising is increasingly becoming a major advertising channel for several firms.

The GSP auction described above differs from traditional auctions in a number of ways. First, search engines display multiple ads in response to a user query. However, the auction cannot be treated as a multi-unit auction because each ad position is different in the sense that top positions generate more clicks for the same number of ad impressions. Further, the CPC decreases as the rank of an ad increases (i.e., the CPC is higher for top ranked ad than a lower ranked ad). Thus, the advertiser has to trade-off a higher number of clicks attained at a top position against the lower margin per click. Due to this trade-off, it may sometimes be better for an advertiser to underbid and sacrifice a few clicks in order to get a higher margin per click. Indeed, several authors have demonstrated that popular second-price search auctions such as those used by Google and Yahoo are not incentive compatible [4, 27]. Thus, bidding one's true valuation is often suboptimal.

In addition, advertisers have short-term budget constraints which imply that bids cannot be submitted independently for keywords. For example, if the advertiser submits a very high bid for the keyword "car rental" then it may leave a very limited portion of the budget for another keyword. The performance of the keywords may also be interdependent, wherein clicks for one keyword may help generate more searches and clicks for another. Therefore the bids for the thousands of keywords are inextricably linked.

Finally, considerable uncertainty exists in the sponsored search environment. For example, the number of queries for "car rental Greece" on any given day is stochastic and is a function of the season, special events, and a variety of other unknown factors. Similarly, consumer click behavior cannot be precisely predicted and the bids of competitors are also unknown due to the sealed bid nature of the auction. The stochasticity in query arrival, consumer click behavior and competitors' bids imply that the number of clicks and total cost associated with any bid are all stochastic.

All these factors - namely the incentive incompatibility of the auction, budget constraints, large portfolio of keywords with interdependent performance and uncertainty in the decision environment - make the advertiser's problem of bidding in sponsored search a non-trivial optimization problem. In this chapter, we formulate and propose a solution to the advertiser's decision problem.

5.2 Related Work

Mechanism design. The main volume of literature relevant to ad auctions is focused toward game theoretic aspects [65] and the design of an efficient ad auction mechanism to improve user experience [3].

Budget Optimization. Assuming the ad auction mechanism of a search engine, the main issue that the advertisers are facing is to decide their bidding strategy and how they are going to split their budget among the keywords of their campaign. There have been various attempts to solve the budget optimization problem, some of which are based on heuristics, some calculate approximations using linear programming variations, and others take a more statistical and stochastic approach.

Even Dar et al. [29] present their approach of maximizing profit, using a linear programming (LP) based polynomial-time algorithm. To deal with the NP-hardness of the problem, they propose a constant-factor approximation when the optimal profit significantly exceeds the cost. It is based on rounding a natural LP formulation of the problem.

Szymanski and Lee [94] discuss how advertisers, by considering minimum return on investment (ROI), change their bidding and, consequently the auctioneer's revenue in sponsored search advertisement auctions. Borgs et al. [10] propose a bidding heuristic that is based on equalizing the marginal ROI across all keywords, so they change each keyword bid based on the ROI performance of the previous day. Their system converges to its market equilibrium in the case of the first price mechanism with a single slot when everybody adopts the proposed perturbed bid solution. Another interesting heuristic that uses a simple uniform strategy can be found in the work of Feldman et al. [30].

Rusmevichientong and Williamson [86] develop an adaptive algorithm that learns the proportions of clicks for different keywords by bidding on different prefix solutions, and eventually converges to near-optimal profits, assuming that various parameters are concentrated around their means. Their model ignores however the multi-slot context. Muthukrishnan et al. [78] consider stochastic algorithms that attempt to solve the problem in advance, and not by adaptive learning as in [86], and work for pre-specified probability distributions of keyword clicks. The authors focus on a single slot auction and find that prefix bidding strategies that bid on the cheapest keywords work well in many cases. However, they find that the strategies for single slot auctions do not always extend to multi-slot auctions and that many cases are NP hard.

Zhou et al. [113] model the problem of advertisers winning an ad slot for one keyword they bid upon as an online multiple-choice knapsack problem. A genetic algorithm approach for solving this kind of problem can be found in [111]. Zhou and Naroditskiy [112] continue the work of [113] modeling budget-constrained keyword bidding as a stochastic multiple-choice knapsack problem. Their algorithm selects keywords based on a threshold function which can be built and updated using historical data. It employs distributional information about prices and tries to solve the bidding problem with multiple ad-position, keywords, and time periods.

The problem of finding a near-optimal bidding strategy has been also approached by using autonomous agents. The TAC Ad Auctions (TAC/AA) game investigates complex strategic issues found in real sponsored search auctions through a simulation of the general auction process [49]. This simulation makes some simplistic assumptions about the ad auction process in order to conduct properly the challenge. In [76] the authors use a genetically evolved strategy that takes into account the position obtained on the exact previous simulated day.

A good contrast between the algorithmic strategies and those of the more simplistic heuristics of the advertisers can be found in the work of Abhishek et al. [2], while at the same time the authors propose an analytical model to compute the optimal bids for keywords in an advertiser's portfolio. However, they make the assumption that consumer click behavior and competitor bidding behavior are independent and identically distributed across ad impressions.

CTR Estimation. Apart from the budget optimization problem and the proposal of an optimal bidding in sponsored search, many recent approaches explore from a more theoretical point of view the critical problem of designing a learning mechanism able to estimate the CTRs [41]. In [39] the authors analyze the issue and at the same time focus on implementing a truthful mechanism with a revenue loss as small as possible compared to an optimal mechanism designed with the true CTRs. In [24] the authors study the multi-armed bandit problems with budget constraint and variable costs. In this setting, pulling an arm will receive a random reward together with a random cost, and the objective of an algorithm is to pull a sequence of arms in order to maximize the expected total reward with the costs of pulling those arms complying with a budget constraint.

5.3 Budget Optimization Problem Definition

The most challenging issue in the managing process of an advertising campaign is the *Budget Optimization* for the multiple keywords of the campaign. We consider the problem as follows: Assuming a limited budget B, we aim to find the combination of keywords with bids that maximizes the campaign profit. In particular, we are looking for a set of keywords $k \in K$ (K is the set of all possible relevant keywords), and their bids $b \in \mathbb{R}_{\geq 0}$ with

$$\sum_{k \in K} w_k(k, b) \leq B \tag{5.1}$$

where w_k is the actual charge when the bidding value is b on keyword k (otherwise called weight) that produce:

$$\max \sum_{k \in K} v_k(k, b) \tag{5.2}$$

where v_k is the function that computes the expected profit of keyword k (value) assuming of bidding value b. We also consider that for any given k with $b = 0$, $b_k = 0 \Rightarrow w_k(k, b) = 0$ and $v_k(k, b) = 0$. A zero bid actually means that we choose not to bid on the particular keyword, so there is no cost or profit produced. In the following sections, we present our approach of finding the best combination of keywords and bids that produce maximum profit. Profit can be either monetary profit from product sales or generated traffic (clicks on ads) for the advertiser's website. We define the above concepts as follows:

Definition 1. *(Weight and Cost)* The cost of a keyword k for a given bid b is the product of expected number of clicks and the average cost per click.

$$w(k,b) = \overline{CPC}(k,b) * Clicks(k,b) \tag{5.3}$$

In Definition 1 $Clicks(k,b) = CTR(k,b) * Impr(k,b)$, CTR = Click-through-rate, $Impr$ = Impressions, $\overline{CPC}$ = Average-cost-per-click

Definition 2. *(Value for maximum monetary profit)* The profit from each keyword-bid combination comes from subtracting the cost of clicks, which is the cost of advertisement, from the revenue of sales.

$$v(k,b) = Revenue(k) * CR(k,b) * Clicks(k,b) - w(k,b) \tag{5.4}$$

In Definition 2 $CR(k,b) * Clicks(k,b)$ is the total conversions (sales) that we expect to have and $Revenue(k) * CR(k,b) * Clicks(k,b)$ is the revenue expected for (k,b), CR = Conversion-rate, $Revenue$ = Revenue-per-conversion.

Definition 3. *(Value for maximum traffic)* When we are interested in maximizing the traffic led to a website, the only valuable measure is the amount of clicks that are generated from keywords.

$$v(k,b) = Clicks(k,b) \tag{5.5}$$

5.3.1 Multiple-choice knapsack problem formulation

In the online advertising campaign, the advertiser plays the role of an investor. The capital is the total budget B for the period that the campaign is active. The profit from the conversions or clicks for each investment is represented as v. The cost that the advertiser is finally charged for a specific investment is w . Each investment is represented by a candidate item x which is a pair (k,b) where

k is the keyword and b the bid that the advertiser initially sets as maximum CPC for the specific keyword. The advertiser has j options of (k, b) candidate pairs for each investment, but he must select only one pair per investment for his final proposal, because for a particular keyword k in the auction process, he can set only one bid. The total number N of the final chosen investments must be equal to the r available keywords of the campaign. This is a Multiple-Choice Knapsack Problem (MCKP). MCKP is a 0-1 knapsack problem in which a partition $N_1 \cdots N_r$ of the item set N is given and it is required that exactly one item per subset is selected. Formally for our problem, the objective is to

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{r} \sum_{j \in N_i} v_{ij} x_{ij} \\ \text{subject to} \quad & \sum_{i=1}^{r} \sum_{j \in N_i} w_{ij} x_{ij} \leq B \end{aligned} \tag{5.6}$$

$$\begin{gathered} \text{with} \sum_{j \in N_i} x_{ij} = 1, \text{ for all } 1 \leq i \leq r \\ \text{and } x_{ij} \in \{0, 1\}, \text{ for all } 1 \leq i \leq r \text{ and all } j \in N_i \end{gathered} \tag{5.7}$$

The above imply that only one bid option is going to be selected for each keyword.

The optimal solution of the MCKP will indicate the best possible choice of keyword-bid options. Our approach was to model this combinatorial optimization problem in a certain way where we can also formulate it as a genetic algorithm (GA) process. In MCKP, the goal is to find for each keyword the option that maximizes the achieved profit. In GA, different chromosomes represent different instances of candidate items and the goal is to find the fittest chromosomes. As we will describe later, a GA finds approximately the proper options of MCKP for profit maximization. This process aims to collect proper statistics from previous time periods and keep only the most profitable options for the next time period. This problem formulation, as we can see in Figure 5.1, is different from the approach that we have seen in [112] and [113], as in our method we focus on the clicks each keyword gains, rather than use MCKP to model the ad auction policy, where each advertiser can select to win at most one ad slot for each keyword. In our formulation, items are options of keyword-bid pairs along with their profit v and cost w, while chromosome $\equiv$ set of selected items.

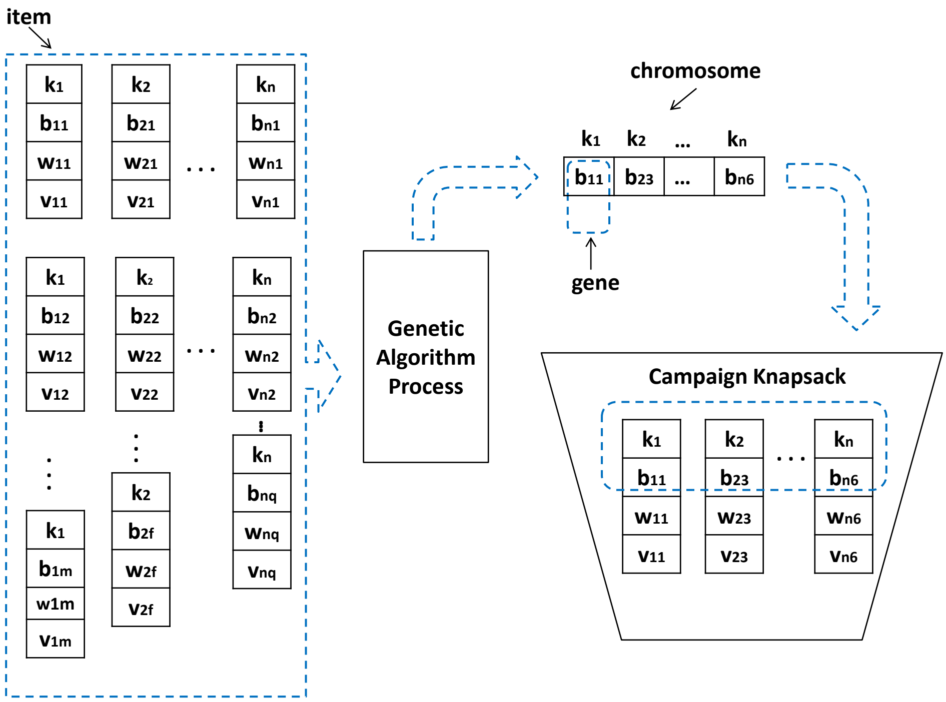


FIGURE 5.1: Mapping of campaign system to the MCKP

5.3.2 Genetic Algorithm Advantages

Multiple-choice knapsack [71] is a known NP-Complete problem, although some solutions for approximate optima in (pseudo-)polynomial time have been found. The approach we take is to capitalize on genetic algorithms [25] that have also polynomial complexity and are used in a variety of global optimization [74], [101] problems. GAs find optima in certain search spaces and are able to combine *exploration*, the process of discovering possible solutions in search spaces, and *exploitation*, the process of using the knowledge of past solutions (past generations) to the benefit of a new more advanced solution. GA finds optimal solutions or near-optimum, since it is an approximation method, like any other polynomial time method that exists. Deterministic methods result in the same approximate solution in each run, thus making it difficult to collect data for many keywords. This is because this method will use those keywords that were chosen repeatedly in the past.On the other hand, the solution of a genetic algorithm may vary, resulting in a different near-optimum solution in each run. This trait is an advantage, as we do not want our method to have obsessions with certain solutions, thus choosing persistently certain keywords. This kind of flexibility, allows our system to discover faster whether keywords are performing better or worse than they did in the past. Therefore, as ad campaign parameters change, something frequent in the case of ad auctions, a deterministic method will adapt much slower than a genetic algorithm.

5.4 Bidding Strategy

The goal is to create a population of candidate solutions (called chromosomes). In each successive generation, a new population of chromosomes is produced by combining (with a procedure called crossover) pairs of chromosomes of the last generation to create new chromosomes (reproduction). The best chromosomes have a better chance to reproduce in the next generation (survival of the fittest), ensuring that each generation is improving. *Selection* is the process of finding the fittest chromosomes to become the parents of the next generation. For this purpose, there are fitness-proportionate techniques such as *Weighted Roulette Wheel Selection* (Weighted RWS) and *Stochastic Universal Sampling* (SUS). These methods make sure that, if a chromosome has a strong fitness, it will have proportionately high probability of reproducing. Moreover, we make sure that a (small) proportion of the fittest chromosomes pass directly to the next generation. This action is called *elitism* and its purpose is to prevent loosing the few best found solutions, increasing the performance of the genetic algorithm. The process of combining two chromosomes is called *crossover*. Every time, two offsprings are produced by two parents and the parents are replaced. The first offspring takes a part of each parent while the other obtains the remaining part of the parents. We want our genetic algorithm to avoid falling in local optima. Thus, the concept of mutation is applied on the chromosomes after the crossover process. Mutation changes the new offspring by altering, with a small probability, the value of their genes increasing the chance for reaching to the global optimum. The process of generating new populations terminates, usually, when $\sim 90\%$ of the chromosomes have the same fitness value or the highest ranking solution's fitness has reached a plateau, i.e., successive iterations no longer produce better results. Alternatively termination occurs in case the number of generations is greater than a certain limit.

5.4.1 Genetic Algorithm Strategy Steps

In this part, we present the main process of our methodology implementation. First, we describe the primary steps for initializing system parameters. We must define a default initial bid for all keywords that are going to be tested, so given a specific variable from Google AdWords, we set
$b_{initial} \leftarrow maxEstimatedFirstPageBid$. This variable is an estimated bid amount offered by the auctioneer that approximates the bid needed for the ad to reach the first page of Google search results when a search query exactly matches the selected keyword. Next, we define time for task periods (e.g., 2 days) and adgroups for each landing page along with their keywords and text ads. In Algorithm 2, we describe the general form of training periods to test campaign and adgroup settings in order to collect proper statistics. The genetic algorithm step is the implementation of the optimization process. Finally, in each testing phase after optimization, we follow the same process of the first training periods but we also pause previous keywords that are not selected by the optimization module.

The reason for different training periods with only a small amount of testing keywords maintained is that, due to a limited daily budget for our experiments, we do not want to exhaust the budget without having tested many keyword options.

Optimization step

In the genetic algorithm, the bids for each keyword that are available to choose from are ones that have been tested and we have kept statistics on. It is not possible to have full information about the performance of a keyword that has not been tested at some point. Important performance criteria for our method are *click-through rate*, *impressions*, *average cost-per-click* and *conversion-rate*. Once we have collected the proper statistics, we are ready to apply our genetic algorithm for optimization.

Algorithm 2: Training Period

```
Input: Settings of Adgroups
Output: Collected statistics
Let t be the number of task periods
Let S_G ⊂ N_G, where N_G are all the candidate keywords of AdGroup G
▷ Make a subset of |S| keywords for testing for each AdGroup G
▷ |G| is the total number of AdGroups
|S_G| ← |N_G|/t
forall the g ∈ |G| do
    add(AdGroup[g].getMostRelevantKeywords(|S_G|), keyword)
end
|M| ← |G| ∗ |S|
forall the μ ∈ |M| do
    setBid(b_initial, keyword[μ])
    activate(keyword[μ])
end
if not firstPeriod then
    chooseRandom(keyword)
    forall the μ ∈ |M| do
        if choosed(keyword[μ]) then
            bidNew[μ] = bidPrevious[μ] ± bidPrevious[μ] ∗ 50%
    end
    forall the μ ∈ |M| do
        ▷ Do not test again other keywords that received clicks
        if notChoosed(keyword[μ]) and receivedClicks(keyword[μ]) then
            pause(keyword[μ])
    end
while taskPeriod > 0 do
    forall the μ ∈ |M| do
        stat[μ] = collect(impressions[μ] ∩ clicks[μ] ∩ conversions[μ] ∩
        averageCPC[μ])
        add(stat[μ], statistics)
    end
    taskPeriod ← taskPeriod − 1
end
return statistics
```

Genetic Algorithm Representation

1. Start Generate random population of m $(m = 40)$ chromosomes

Chromosome Representation

- For the budget optimization problem, each chromosome consists of N genes, N being the number of available keywords

  Each gene has a value of the bid index that is selected for the specific keyword

- Table 5.1 shows a chromosome that has selected the second bid for keyword k_1 and zero bid (value 0) for keyword k_2
- Table 5.2 shows that the second bid ($bidIndex = 2$) for k_1 is the actual bid value of \$0.60
- Table 5.3 shows that this bid has a cost of \$16.2 and a positive profit of \$1.40. If a keyword is not selected ($bidIndex = 0$), like k_2 in Table 5.1, it produces zero cost and profit

2. Fitness Fitness Function Evaluation

- The fitness function is the expected total profit for the bids selected in the chromosome genes.

$$\text{Chromosome Fitness} = \sum v(k_i, b_i) \tag{5.8}$$

  Fitness function resembles the objective function of the knapsack problem. It can be easily computed since we have pre-computed all the costs and profits of the bids for every keyword, as shown in Table 5.3

  Evaluate the fitness function of each chromosome in the population. Take into consideration *actual* or *predicted* values

- When a chromosome is generated, it has to pass the $\sum w(k_i, b_i) \leq B$ condition, otherwise randomly selected genes of the chromosome will be set to 0 until the condition is met

3. New Population Create a new population by repeating the following steps until the new population is complete:

a. Selection Select two parent chromosomes from a population according to their fitness (Weighted RWS). The best chromosomes are the ones with the highest values of the fitness function

b. Crossover With a crossover probability, cross over the parents to form a new offspring (children). If no crossover was performed, offspring is an exact copy of parents

c. Mutation With a mutation probability ($\sim 0.1\%$) mutate new offspring at each locus (position in chromosome)

d. Accepting Place new offspring in a new population

4. Replace Use new generated population for a further run of algorithm

5. Test End Condition

- Since we don't know what the best answer is going to be, we just evolve the max number of times *(MaxAllowedEvolutions = 3000)*

  If the end condition is satisfied, **stop**, and return the best solution in current population

Loop Repeat from step **2**

After a period of testing and collecting statistics, the budget optimization task is ready to run again. This process continues executing until the last day of our campaign.

	k_1	k_2	k_3		k_N
$bidIndex$	2	0	3	...	1
$valueRange$	[0-4]	[0-3]	[0-3]		[0-2]

TABLE 5.1: Example of chromosome representation and the values of its genes

5.4.2 Impressions Prediction

We wanted also to examine if a certain type of campaign behavior prediction could be helpful to our system. Although as we mentioned earlier there is a

bidIndex	k_1	k_2	k_3	...	k_N
1	$ 0.50	$ 0.90	$ 0.45	...	$ 0.55
2	$ 0.60	$ 1.10	$ 0.55		$ 0.70
3	$ 0.70	$ 1.30	$ 0.65	...	
4	$ 0.80				

TABLE 5.2: Bid matrix example. For k_1, we have a value range of [0-4] of the bidIndex

k_1			...	k_N		
bidIndex	**w(k,b)**	**v(k,b)**	...	**bidIndex**	**w(k,b)**	**v(k,b)**
1	$ 14.5	$ 1.5	...	1	$ 11.0	$ 1.5
2	$ 16.2	$ 1.4	...	2	$ 16.5	$ 1.9
3	$ 18.1	$ 0.3				
4	$ 19.8	$ 0.5				

TABLE 5.3: Example of expected costs and profits for each different (k, b) pair

general stochasticity in the auction and competitors' behavior, we aim to integrate as alternative options, an approximate prediction of new values in the next auction instead of using only past ones from historical data.

Clicks, click-through rate, and conversion rate are parameters that are more dependent to inner factors of the advertiser's choices such as the quality and relevance of the selected keywords and ad-texts for the product promotion. However, this is not exactly the case for the impressions that a user query generates. The impressions fluctuate primarily and because of other factors external to the keyword-bid combination. Consequently, we need a means to predict or at least make a good estimation of how many impressions a keyword will receive matched with a specific bid, knowing:

1. Past received clicks for various selected (k, b) combinations
2. Current average user searches for a query similar to this keyword
3. Current will of competition of all the other bidders upon this specific keyword

The idea is to use past results of keyword behavior in a model that can capture externalities of the ad auctions and predict current or future behavior. Google AdWords provides information such as Global Monthly Searches (GMS) and

Competition of a keyword which are factors that affect the number of impressions of a keyword and, at the same time, are independent of a particular AdWords Account.

Past data of all keywords with known Impressions have the following form:

$[Clicks(k_1, b_1), GMS(k_1), Competition(k_1)] \;\rightarrow\; Impressions(k_1, b_1)$
$[Clicks(k_2, b_2), GMS(k_2), Competition(k_2)] \;\rightarrow\; Impressions(k_2, b_2)$
$\ldots \quad \ldots$
$[Clicks(k_n, b_n), GMS(k_n), Competition(k_n)] \;\rightarrow\; Impressions(k_n, b_n)$

thus, we aim to predict the impressions of another keyword $-$ bid (k_i, b_i):
$[Clicks(k_i, b_i), GMS(k_i), Competition(k_i)] \;?\rightarrow\; Impressions(k_i, b_i)$

After prediction of impressions of all keyword-bid combinations is carried out, new values for clicks and conversions can be computed. A good estimation of impressions may result in a good cost and profit estimation, and can possibly lead to an improved budget optimization. To perform impressions prediction, we choose multiple linear regression [75]. This method assumes the existence of linear correlation between the dependent variable y (Impressions) and the independent variables (in our case x_1 = Clicks, x_2 = GMS, x_3 = Competition). So, we need to find the best coefficients that show the relationship between y and x_i. The goal is to be able to calculate a new value of y out of the independent variables and the coefficients.

$$y' = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + ... + \theta_k x_k \tag{5.9}$$

In our case, we have 3 independent variables:

$$y' = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 \tag{5.10}$$

The regression model is fitted with the least squares [54] approach. The sum of square residuals is considered to be the error (e) when comparing the y with y':

$$e = \sum_{i=1}^{N} (y_i - y'_i)^2 \tag{5.11}$$

N is the amount of all available records. In the end, the chosen coefficients θ_i must minimize the error produced by prediction to have the best fit of our model.

The result from this process is an alternative input to the genetic algorithm with different calculated statistics in order to study if using impressions prediction would achieve better campaign performance.

5.5 Experimental Evaluation

We present here several experiments and data analysis that we have conducted in order to study the performance of the proposed optimization methodology.

5.5.1 Evaluation Data

We use the historical data of a large scale AdWords Campaign of a web site in the area of car rental. We selected from the collected data all the campaigns and adgroups that promote "car rental in Crete". The data collected derive from the period May 2009 to November 2010, during which the campaign was very active the majority of the time, generating traffic and sales for the car rental website. With the retrieved data of Google AdWords keyword statistics and sales for the car renting business, we get sufficient data for 39 weeks of this large scale campaign to perform tests on the impressions prediction and budget optimization modules. The final form of the integrated statistics table contains the following features: *{Campaign, Adgroup, Week, Keyword, MaxCPC, Impressions, Clicks, Conversions, CTR, Avg CPC, Cost, Profit, Quality Score, FirstPageCPC, Avg Position, Avg CPM}*. For the impressions prediction module of the system, we need to have for every keyword the "Global Monthly Searches" and the "Competition" values, retrieved using the AdWords API.

Our budget optimization system provides two options; to optimize budget for maximum traffic or for maximum profit. Additionally, we can use original or predicted impressions. These options give us four basic testing scenarios:

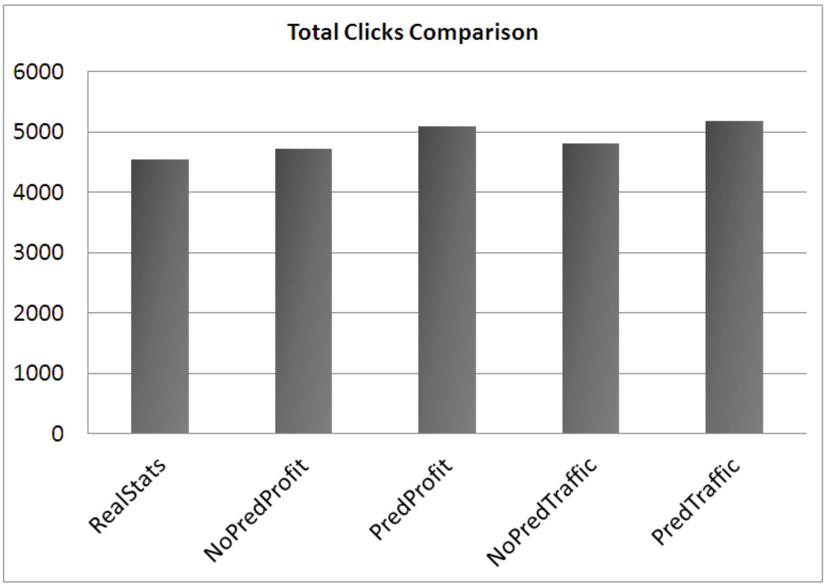


FIGURE 5.2: Traffic weekly performance BudgetOptimization vs. RealStats

1. Budget Optimization for Profit with No Prediction (NoPredProfit)
2. Budget Optimization for Traffic with No Prediction (NoPredTraffic)
3. Budget Optimization for Profit With Prediction (PredProfit)
4. Budget Optimization for Traffic With Prediction (PredTraffic)

5.5.2 Genetic algorithm performance on finding the best solutions for MCKP

In this experiment, we apply the genetic algorithm to evaluate the hypothesis of choosing the optimal keyword-bid combination of each week. The input weekly budget for our scenarios is the corresponding actual weekly cost of the campaign. For each week, the input keyword options for the genetic algorithm are the actual tested keywords and bids for the specific week. Each scenario output is the average result of five executions of the genetic algorithm. In Figures 5.2 and 5.3 we present the results for total traffic and profit comparison, where we notice that our method finds in total the most profitable keywords for both traffic and profit maximization cases.

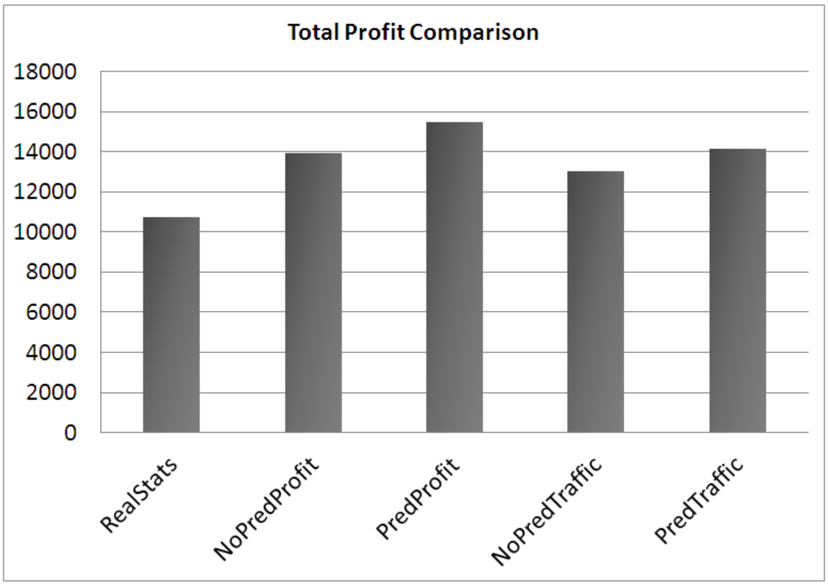


FIGURE 5.3: Profit weekly performance BudgetOptimization vs. RealStats

5.5.3 Genetic algorithm performance on optimizing next week's performance

In this experiment, we test the expected weekly performance of each of our methodology scenarios towards the actual campaign weekly performance. For estimating performance of week i, the genetic algorithm takes into consideration the statistics from weeks 1 to $i-1$, resulting in a "leave-one-out" cross-validation-like process. The training set is the actual statistic set from week 1 to $i-1$ and the testing set is the actual statistic set of week i. For example, the input features for the optimal keywords and bids of the 20th week are the collected statistics from weeks 1 to 19. The purpose of this evaluation is to find solutions that achieve higher weekly performance than the actual one. Each scenario output from the budget optimization process is the average result of 10 executions of the genetic algorithm. The input weekly budget for our scenarios is a bit higher (1-2 euros) than the corresponding actual weekly cost of the campaign, assuming without having the actual information, that on average the budget is not completely depleted. As we present in Figures 5.4 and 5.5, in the case of traffic maximization as the advertising goal, our two methods which use prediction, surpass the real results. In this experiment, the optimization process had started after the 4th week, because the advertiser until the 3rd week had been testing very few keyword options (3-4) and the GA needs more testing data to perform a valid optimization. The important observation

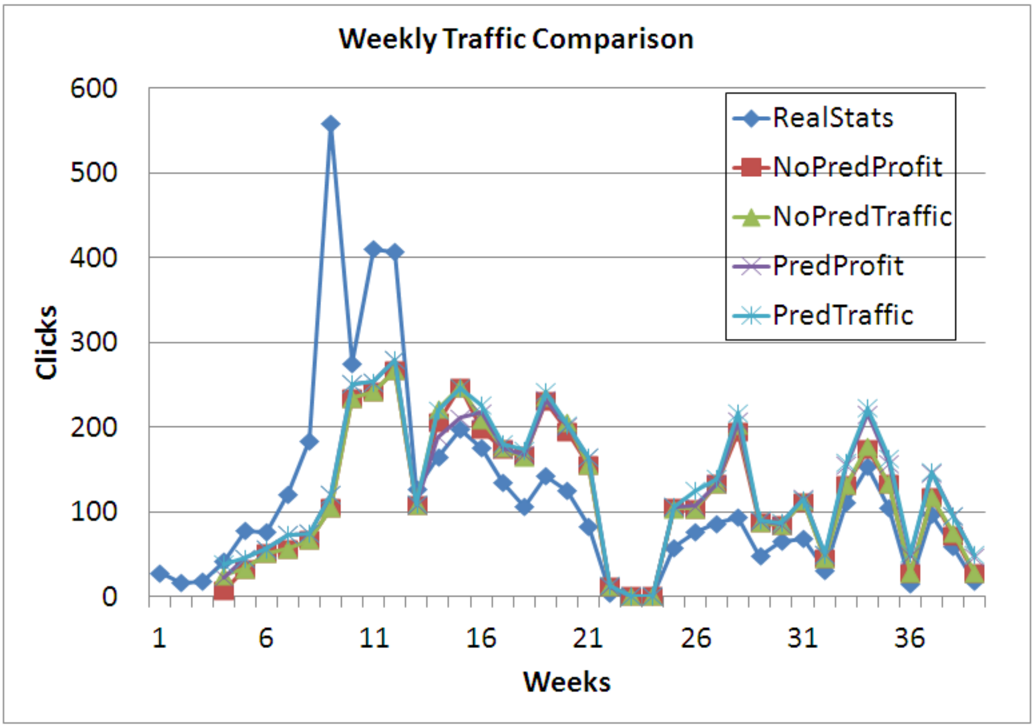


FIGURE 5.4: Traffic weekly comparison for next week's optimization

here compared with the stronger performance of the previous experiment was the use of much older and thus outdated data that did not correspond to valid receiving impressions and clicks in the ith week. The Impressions Prediction module had a major contributed role in the calculation of more up-to-date data because it achieved to capture current external factors and conditions of the ad auction. Thus, the methods that were using prediction outperformed the other ones.

5.5.4 Scenario Comparison

The data used for this experiment on the budget optimization process are the keyword statistics we collected from the car rental website for 39 weeks and the budget to be allocated for the next (hypothetical) week. Since budget optimization is performed with a genetic algorithm -a stochastic method- the result will slightly vary every time it is executed, even with the same input data. So, each scenario (NoPredProfit, NoPredTraffic, PredProfit, PredTraffic) is executed 30 times and the result reported is the average value of 30 executions.

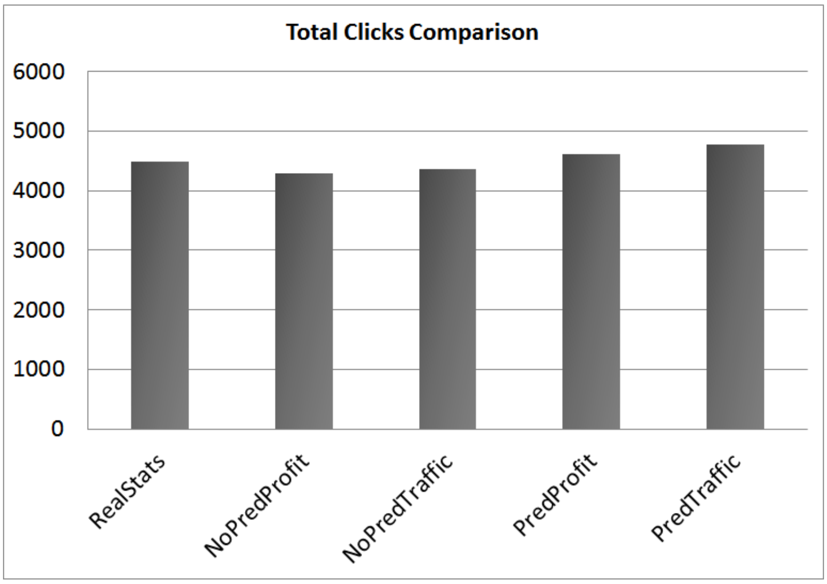


FIGURE 5.5: Total clicks comparison for next week's optimization

The result of every execution of the budget optimization module is an optimal keyword-bid combination that ensures either maximum traffic or maximum profit for a limited budget. In particular, every result of the genetic algorithm application produces the following data:

- *Clicks*: How many clicks is the optimal solution (keyword - bid combination) expected to produce in the following week? This is an estimation, so it is represented with a double instead of an integer value
- *Cost*: How much is it expected to cost in the following week? This value must always be lower or equal to the budget
- *Profit*: How much profit are we expected to make in the following week? The profit is calculated after excluding the advertisement cost, meaning: $Revenue = Cost + Profit$
- *#Keywords Used*: This value counts the number of keywords which were selected in the optimal solution
- *Average Bid*: The average value of the bid (or *MaxCPC*) of every selected keyword of the optimal solution

The above output is the average result solution of the applied budget optimization for a future 40th week of the advertising campaign. This experiment is using a simulation and we make here the assumption that the metrics are computed as if CTR, clicks, costs, and impressions were maintained the same for each (k, b) choice in the future. We first run budget optimization for different values of the available budget. In Table 5.4, we present the average results of 30 executions for the four scenarios with budgets of 50, 100, 200, 400, and 600 units (euros).

Budget = 50	**Clicks**	**Cost**	**Profit**	**#Keywords Used**	**AverageBid**
NoPredProfit	60	49.94	219.51	24	1.49
NoPredTraffic	61	49.93	206.22	23	1.43
PredProfit	82.36	49.90	317.1	16	1.37
PredTraffic	86.51	49.88	274.81	18	1.42
Budget = 100	**Clicks**	**Cost**	**Profit**	**#Keywords Used**	**AverageBid**
NoPredProfit	108	99.93	374.98	25	1.48
NoPredTraffic	109	99.92	356.44	26	1.44
PredProfit	130.80	99.87	467.86	20	1.41
PredTraffic	134.21	99.92	364.53	19	1.46
Budget = 200	**Clicks**	**Cost**	**Profit**	**#Keywords Used**	**AverageBid**
NoPredProfit	197	199.87	621.32	56	1.55
NoPredTraffic	200	199.90	582.21	54	1.50
PredProfit	236.94	199.86	787.63	31	1.42
PredTraffic	248.60	199.85	638.13	32	1.43
Budget = 400	**Clicks**	**Cost**	**Profit**	**#Keywords Used**	**AverageBid**
NoPredProfit	333	389.61	798.90	98	1.61
NoPredTraffic	340	399.92	791.93	102	1.63
PredProfit	425.74	399.82	1313.99	54	1.51
PredTraffic	447.42	399.90	1191.51	45	1.45
Budget = 600	**Clicks**	**Cost**	**Profit**	**#Keywords Used**	**AverageBid**
NoPredProfit	333	389.60	798.90	97	1.61
NoPredTraffic	343	405.16	795.28	107	1.63
PredProfit	607.74	599.84	1645.60	70	1.56
PredTraffic	622.69	599.82	1569.21	68	1.52

TABLE 5.4: Budget optimization evaluation results

We notice the following on all tests: The methods that were using prediction outperform the simple GA ones. Optimization for profit always produces more profit than optimization for traffic, as expected. Optimization for traffic always produces more clicks than optimization for profit, as expected. We notice that the Average Bid increases along with the available budget. This is because the cheaper (cost-efficient) keywords are running out, so we have to use more costly ones. All solutions deplete their budget unless there are no more keywords left or the keywords left are not profitable. In the case of budget=600, when optimizing for traffic without prediction, we reach the limit of how many clicks can be made, and therefore our solution produces the maximum cost (405.16), which is less than the budget (600). This solution also uses all available keyword options (107 in size). In the case of budget=400 and budget=600, when optimizing for profit without prediction, we reach an upper limit of the profit, so the budget is not depleted. Not all keywords are used in this case because not all keywords are profitable. In the cases of small budgets, we notice that optimizing for profit generates almost as much traffic as optimizing for traffic. This could mean that keywords that generate more profit are more relevant, hence they are clicked more often.

5.5.5 Comparison of parallel competing campaigns

In this experiment, we create Google AdWords campaigns for two companies. Client1 is a company that offers web developing solutions (a highly competitive field for online advertising) and Client2 is a company that offers aluminum railing and fencing products. For each company we create one manual and one automated campaign. A human administrator was responsible for the setup of the manual campaign assisted by some baseline changes in the course of the experiment from the optimization tool of AdWords. Each automated campaign is created semi-automatically by our system (the only intervention is the parameter input of daily budget, account credentials, period of active campaign, and keywords).

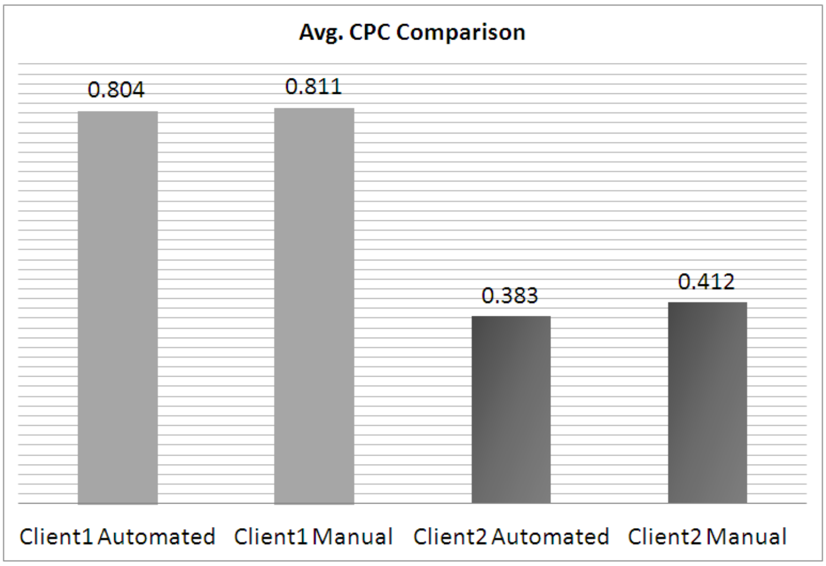


FIGURE 5.6: Automated vs. Manual campaign Avg. CPC

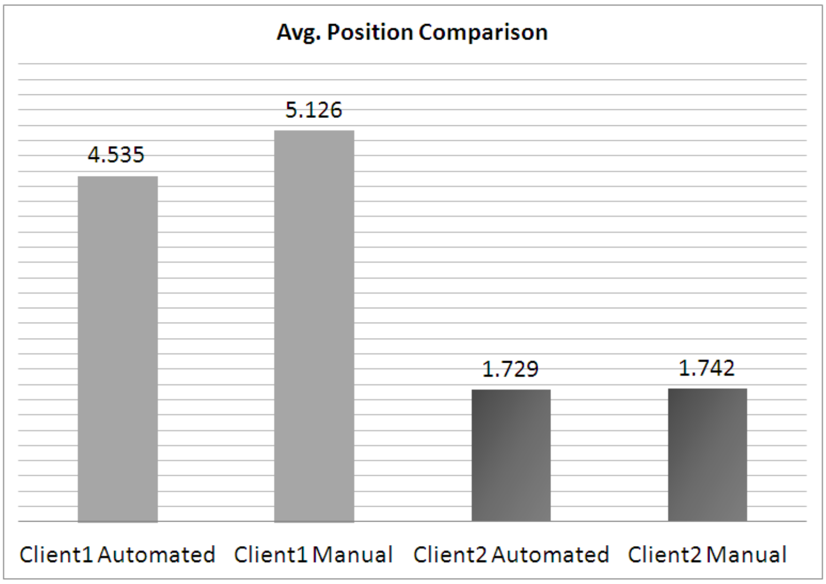


FIGURE 5.7: Automated vs. Manual campaign Avg. Position

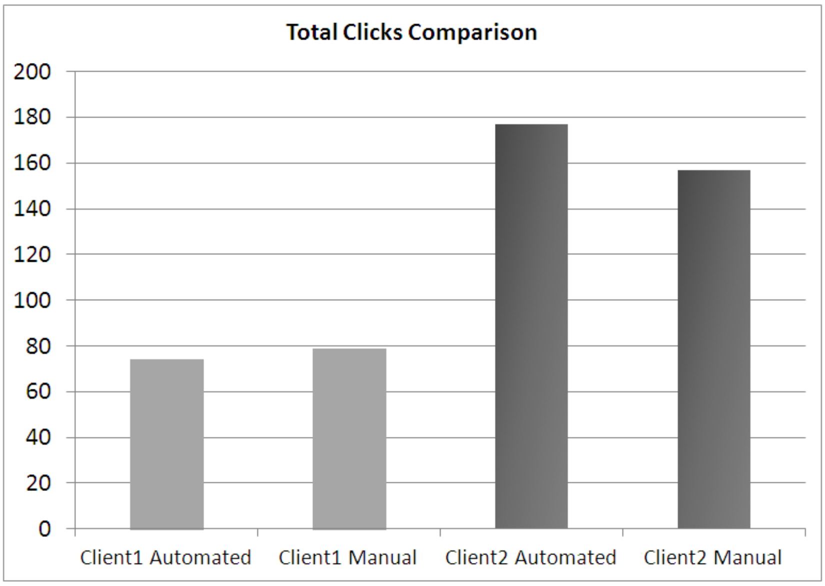


FIGURE 5.8: Total Clicks Comparison for Client1 and Client2

We set our automated campaigns for traffic maximization as the advertising goal. We use for each manual and automated campaign the same keywords and the same budget in order to test only the monitoring and optimization process. In this experiment, we do not use impressions prediction, only the real values case scenario (due to limited budget for further experiments at that time).

In Figures 5.6, 5.7, and 5.8, we present the final results after a period of 17 days. In the case of Client2, the automated campaign achieved higher performance in total traffic than the manual one. In the case of Client1, the automated achieved a slightly lower performance than the manual one. In both cases, the automated campaigns achieved better placement in the advertising slots than the manual ones, as well as lower prices for average cost-per-click.

Chapter 6

The Adomaton Prototype

In this chapter we give an overall presentation of the integrated *Adomaton Prototype* system [96, 99] as well as detailed software design specifications and use cases examples.

We propose the following demarcation of the general framework as we discussed in the previous chapters:

The Keywords and Ads component is responsible for retrieving the most relevant keywords and generating ad creatives based on information taken from the landing pages. The output of this part which generates multiword keywords (n-grams) and automated ad creative recommendations is selected as input feature in the following component.

The Campaign Management component is responsible for initializing, monitoring, and managing the advertising campaigns in the course of time, based on keyword statistics maintained by the system, in the view of optimizing available budget.

The functionality is encapsulated and granted to the end user through an appropriate web interface.

A demonstration of the Adomaton Prototype can be found at http://adomaton.com/

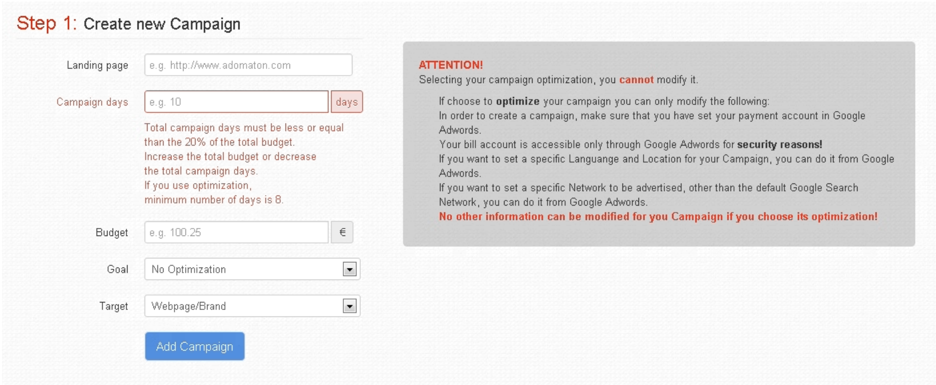


FIGURE 6.1: Adomaton Initialization Settings

6.1 Use Cases

In this section we present the different options for using the Adomaton Prototype. In each case, the user has to have been logged in the system, with the AdWords credentials. Adomaton Servlet starts a new session for this user, assigning all the information needed between requests and responses, to an object, which then is saved in this specific session as attribute. The user selects creating a new AdWords campaign and he is directed to the page that is presented in Figure 6.1.

Afterwards, he fills the information needed for the campaign that the application is going to create, inserts the main URL of the product, service, or brandname that he wants to promote, the period days of the whole campaign, and the budget for the total campaign days. In addition, he selects one of the three system runnable options for the campaign that is going to be activated:

1. *No Optimization*, where the system just uploads automatically the generated keywords, ad-texts, and bids along with their organized structure without continuing to be responsible for an automated optimization strategy
2. *Traffic Optimization*, where the advertiser considers the profit to be the amount of clicks at the ad-texts
3. *Profit Optimization*, where the profit is the actual monetary profit from offline product sales or online conversions to a specific landing page that is defined in a next step

Then, the user selects the advertising target that can be: 1. Website/Brand-name, 2. Product, or 3. Service. This option is useful for the Ad Creative Module in order to generate the proper action phrase.

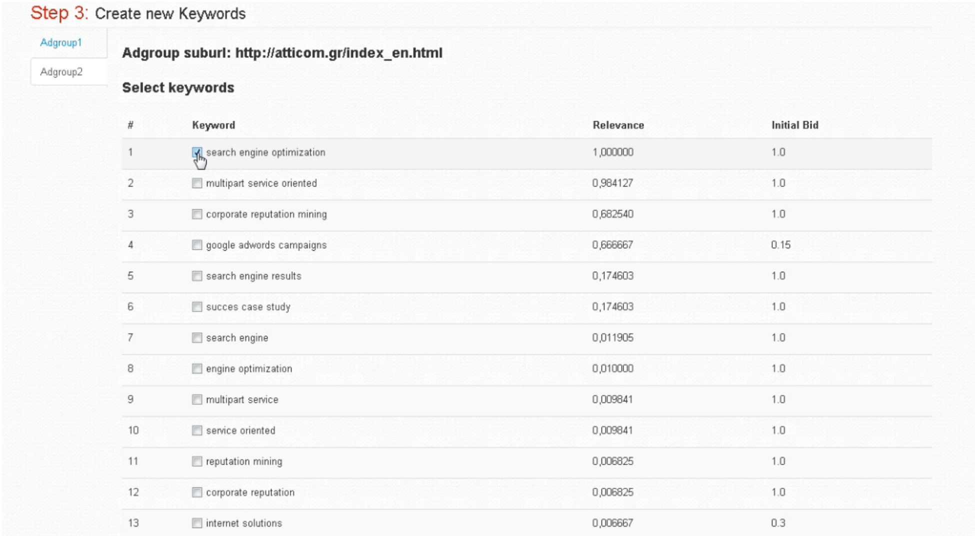
The default option from our system regarding the Google Network where the ad-texts are going to be impressed is only the group of **Search Network**, opting-out the same time from the Display Network group [1].

After the proper settings are inserted, Adomaton Servlet as a second step reads each input information and assigns them to the session object. Depending on the main landing page that has been set by the user, the *Crawler Module* visits the specified webpage and using a web-scraping technique, obtains webpage source information. In this process, the Crawler extracts possible existing sub-landing pages, after validating their availability. Each retrieved sub-landing page is corresponding semantically to an AdGroup in our proposed Campaign Organization. For the case of Profit Optimization, next to each sub-landing page the user can insert a specific monetary profit that he is going to gain from a conversion in this page. This is useful for the Profit Maximization Strategy. The user can select all or a portion of these retrieved landing pages.

As a third step, the user is directed in a page where he must select for each of the previously selected pages the automatically generated keywords. Next to each keyword it is presented to the user a normalized score of its relevance to the AdGroup, as well as an initial bid value. This value is derived from $min(1, maxEstimatedFirstPageCPC)$. This step is presented in Figure 6.2. The estimated first page bid amount approximates the bid needed for the ad to reach the first page of Google search results when a search query exactly matches the keyword. This estimate is based on the Quality Score and current advertiser competition for that keyword. We retrieve this value using the AdWords API. We decided to place an upper bound of 1 due to the fact that a commonly used strategy by the advertisers [2] is to evaluate as a default case the utility of the click at a url to be equal to one monetary unit (e.g., 1 euro or dollar) [2, 10].

[1] We took this decision in our experiments and strategies because choosing to appear also in the Display Network was leading to a large amount of impressions and very few clicks. As a result the values of CTR (Clickthrough rate) were very low (<0.5%) causing in this way low Quality Scores and increased recommended bids for good ad slots. In any case this thesis tackles the Sponsored Search Advertising paradigm, thus our methods were focused on this and not on the Contextual Advertising in which corresponds the option of the Display Network group

[2] http://support.google.com/adwords/bin/answer.py?hl=en&answer=2471184&from=50081&rd=1



Step 3: Create new Keywords

Adgroup1
Adgroup2

Adgroup suburl: http://atticom.gr/index_en.html

Select keywords

#	Keyword	Relevance	Initial Bid
1	search engine optimization	1,000000	1.0
2	multipart service oriented	0,984127	1.0
3	corporate reputation mining	0,682540	1.0
4	google adwords campaigns	0,666667	0.15
5	search engine results	0,174603	1.0
6	succes case study	0,174603	1.0
7	search engine	0,011905	1.0
8	engine optimization	0,010000	1.0
9	multipart service	0,009841	1.0
10	service oriented	0,009841	1.0
11	reputation mining	0,006825	1.0
12	corporate reputation	0,006825	1.0
13	internet solutions	0,006667	0.3

FIGURE 6.2: Automatically generated keywords and recommended bids

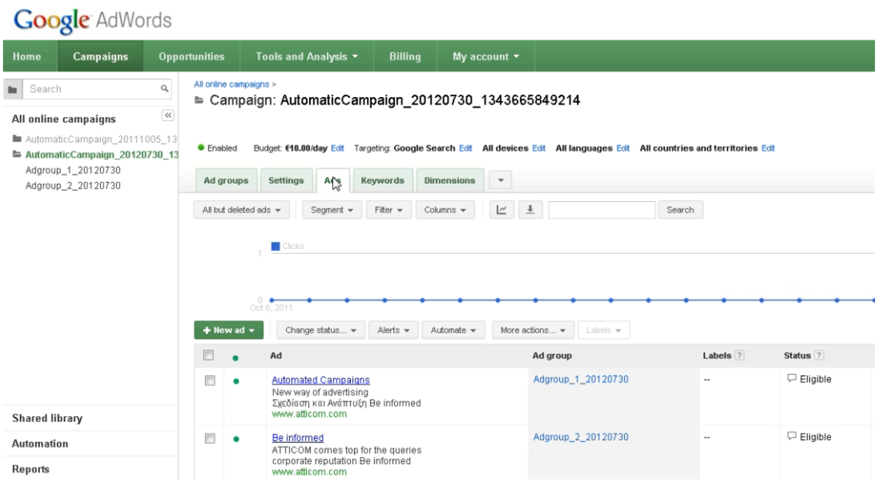


FIGURE 6.3: AdWords Uploaded Settings

Furthermore, the user at the fourth step can select for each sub-landing page the automatically generated advertising text as well as edit each part of it to his needs, before its final upload in the campaign.

Finally, in Figure 6.3 the user can see through the AdWords interface the uploaded settings of his constructed campaign.

6.2 Conceptual Modeling

In the following paragraphs we attempt to describe thoroughly the general concepts, entities and relationships, on which the Adomaton System relies. The

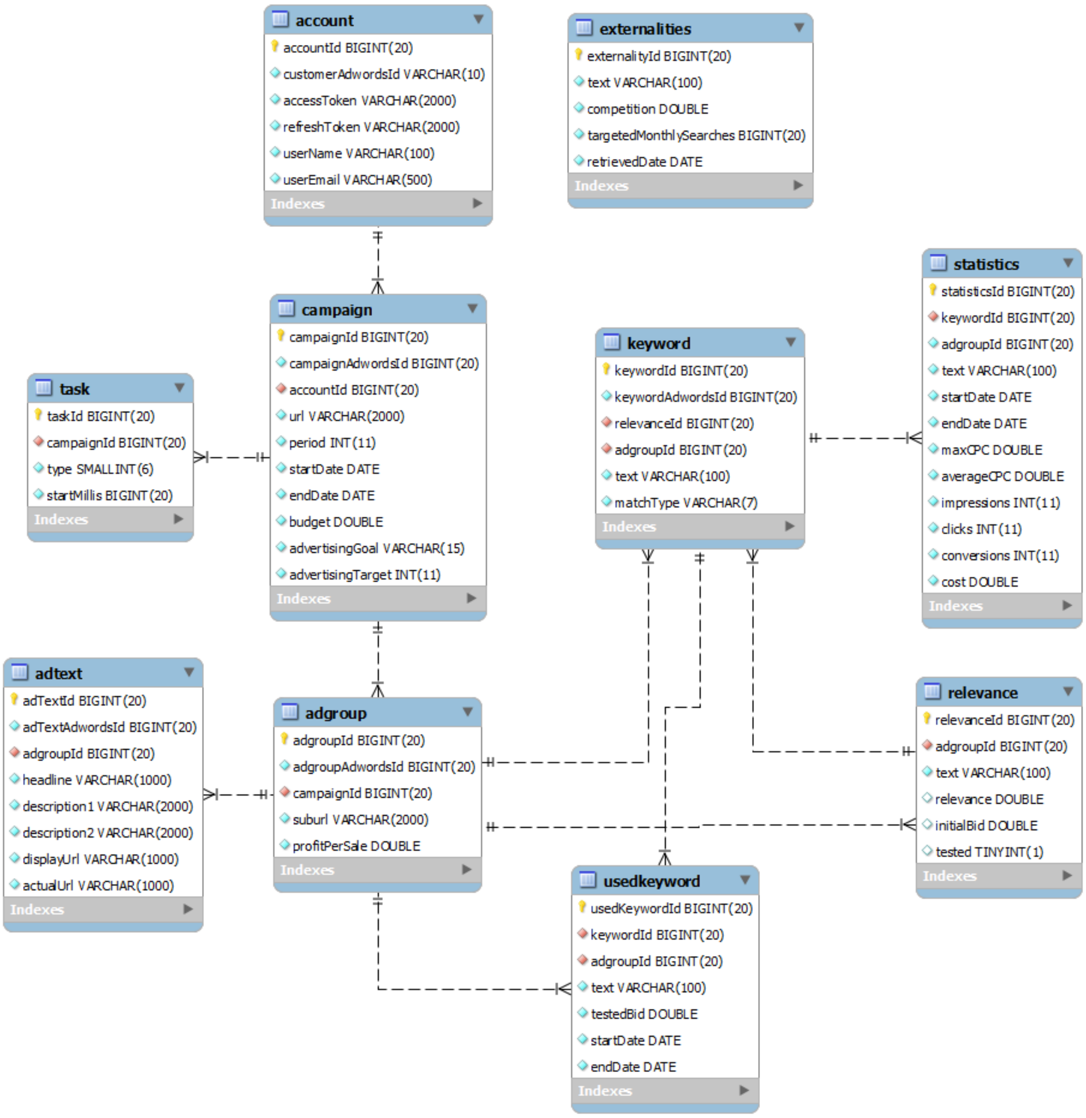


FIGURE 6.4: Adomaton Entity - Relationship Model

database schema and its tables are playing a key-role in the general functionality and representation of the total process.

The database has been designed in order to access faster data and using less computational resources. The way this fact has been achieved is by taking advantage of the rules of the third normal form that has been used for the better organization of the data structure. More specifically our approach has combined these rules, adding some slight information to the tables in order to have faster and less expensive joins, having immediate access to our data.

The diagram in Figure 6.4 provides a visual overview of the Adomaton database schema and the relations between entities.

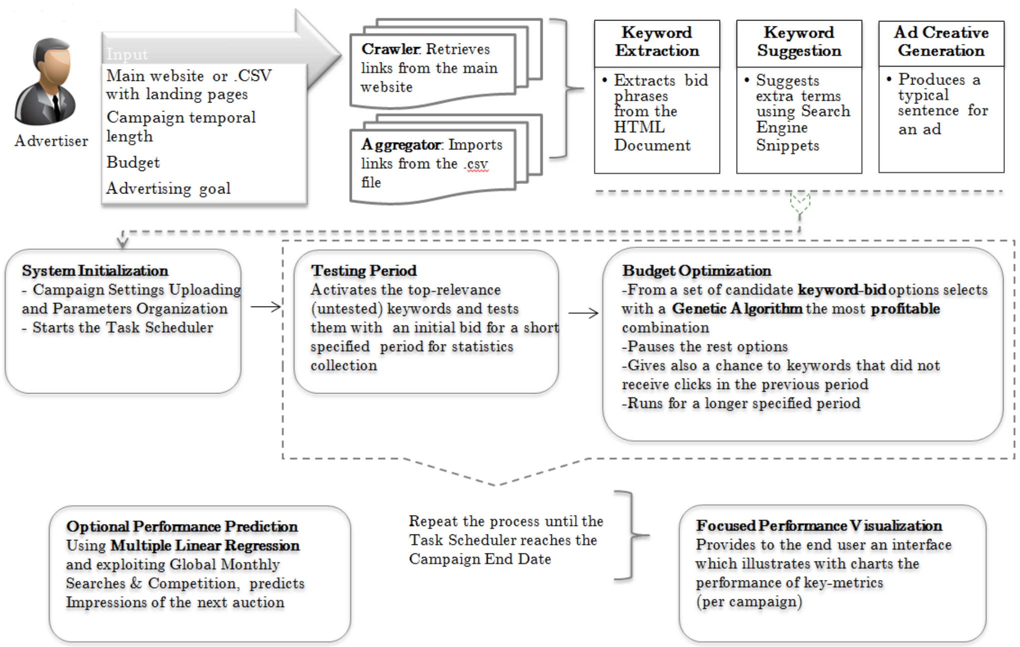


FIGURE 6.5: System Flow

The schema has been depicted as entities with one-to-many relationships. We have the main information that we store, such as information for the Account, the Campaigns that have been created, the AdGroups and the Ad-Texts that correspond to each Campaign, the Keywords that have been produced from our system and their correspondence to each AdGroup. We also have the Statistics and some Externalities, external info for each keyword that has been used as long as each Campaign is active. We get this information from AdWords API, in order to optimize the keyword options of each Campaign. Finally, the Tasks let us know when we have to execute a specific process for each Campaign, while optimizing its budget for better traffic or profit, according to what user has previously chosen. The interested reader may consult appendix A, where we provide details for the table fields of the database and what they represent.

In order to have an overall picture about the system flow, we provide Figure 6.5.

6.3 System Architecture

6.3.1 Design Considerations

The code of this software was completely written in Java using JDK SE 7. For the database engine, MySQL 5.5 Community Server was used. We performed tests on Windows XP, Windows Vista, Windows 7 but it can be also running on Linux servers, as long as there exist a Java, a MySQL installation, and Apache Tomcat as the JSP Server.

The AdomatonServlet is the path to the system's main functionality. Forms that are completed by the user transfer their data to the AdomatonServlet and then the Servlet takes care of the distribution of this kind of data to the internal functionality, while creating a new campaign or visualizing the results of an active and/or completed campaign. When users choose to optimize their Campaign having a specific target, either this is traffic or profit, in the end of the Campaign creation process, they have an initialized Campaign with specific AdGroups, Keywords and Advertisement snippets to be used. Then the system uses for each new Campaign a Scheduler, in order to manage the processes that have to be executed. This Scheduler is actually a thread, which is sleeping until a specified time (in milliseconds). When this specified time comes, it wakes up to read from the database which task is next in line to be executed -if there is one- and activates the scheduled process. In this way, we achieve concurrency for many users through the Servlet that handles multiple http requests from multiple accounts, as well as concurrency for many campaigns through the responsible thread for each campaign.

In the following paragraphs we describe the role of some key classes.

campaign.budgetoptimization For each keyword-bid combination (k, b), an evaluation of the cost and profit must be computed based on the keyword performance information kept in the database. If a bid gives negative expected profit, it must be erased from the data because it cannot contribute to a maximum profit solution. Optionally, instead of directly using the value of keyword impressions to compute cost and profit, prediction can used. The predicted impressions will have an effect on expected clicks that in turn affect cost and profit. The problem is then modeled into chromosomes and the fittest chromosome is finally selected by the genetic algorithm after several generations of breeding using the methods of crossover, mutation, and elitism. A final list of keyword-bid pairs (k, b) is produced from the fittest chromosome. These keyword-bid pairs will form the new bidding strategy, which must be set in the AdWords account of the advertiser using the AdWords API. Evolving the population the max number of times (we set the number of maximum allowed evolutions to 3000) and setting the initial population size of chromosomes to 40, help us avoid premature convergence, which was considered as the main problem of GA theory [85].

campaign.campaigncreation Initialization of all the procedures referenced by the AdomatonServlet. It contains the method startScheduler that saves the initial tasks according to the System Mode/Goal and in parallel starts the Scheduler.

campaign.prediction This is the module that is responsible for performing impressions prediction using the past statistics of keywords and targeted information taken from AdWords, such as the Global Monthly Searches and the level of Competition that exists for a given keyword. Before budget optimization, optionally, we can use this module in order to refine the statistics used by applying impressions prediction. A regression model finds the relationship between the impressions and other variables, such as clicks, global monthly searches, and competition. It then re-computes the value of impressions for each keyword-bid combination using the independent variables. The values of Global Monthly Searches and Competition must be computed from the AdWords API.

campaign.scheduler Since ad campaigns run for weeks, months, or even years, a scheduler is important for the system in order to ensure a task start and end. If a scheduler does not exist, the system cannot store proper statistics and allocate resources properly. It calls SystemTasks methods according to the tasks that are next in line to be executed.

campaign.statistics The database must store all the statistical information collected for the keywords to be able to track their performance. All metrics about keywords are necessary for the budget optimization module function. Both the Genetic Algorithm and the Prediction modules make extensive use of the statistical information of the keywords. There are useful also to the Campaign Performance Visualization Module.

campaign.googleapiwrapper For the sake of programming sanity, many tasks must be wrapped to simpler functions, saving effort in typing, code readability and making the software less error prone. This class is also crucial in order to perform bulk requests (i.e., many operations in one request) as well as having the role of an integrated communicator with the AdWords API in order to migrate more easily to any API changes.

6.3.2 External Tools and Libraries

We enlist here the components and auxiliary libraries that assist to the development and functionality of the Adomaton system.

- Twitter Bootstrap[3]: We designed the web interface of the system based on its clean and beautiful template as well as its CSS Tools.
- HighCharts Charting Library[4]: We used Highcharts as the charting library in the Campaign Performance Visualization Interface.
- Jsoup Java HTML Parser[5]: We used the Jsoup Parser in order to parse and preprocess the landing pages, before applying the keyword and ad creative generation methods.
- Apache Lucene[6]: We used it in order to store as "documents" the snippets from the Search Engine Result Page in the Ad Creative Generation Module.
- Google JSON/Atom Custom Search API[7]: We used it to retrieve in a formal way the search engine result snippets.

[3]http://twitter.github.com/bootstrap
[4]http://www.highcharts.com/
[5]http://jsoup.org/
[6]http://lucene.apache.org/core/
[7]https://developers.google.com/custom-search/v1/overview

- Classifier4J Java Library[8]: It is a Java library designed to do text classification. It comes with an implementation of a Bayesian classifier, and includes also a text summary facility useful for the baseline variation of Ad Creative Generation Process.

- JGAP Genetic Algorithms Java Package[9]: We used its genetic mechanisms for our budget optimization/ bidding strategy (as we have formed it into a Genetic Algorithm Task).

- Flanagan's Java Scientific Library[10]: We used it in order to develop the Linear Regression Prediction Module.

- Google AdWords API[11]: The Google AdWords API lets developers build applications that interact directly with the AdWords platform. It is the main component for our communication with the Google AdWords Platform.

- Google Ads APIs Client Library for Java[12]: It is a Java client library for the SOAP-Based Ads API of AdWords. This library makes it easier to develop our Adomaton Java client to programmatically access all relevant information from the AdWords Platform.

- GrammAds: We have developed also a tool that can be used as an individual component for the process of *Keyword Generation* as we have proposed in Chapter 3. It can be found in http://prototypes-db-net.aueb.gr:8080/GrammAdsDemoDBNET/.

In Figure 6.6 we present the overall System Architecture.

6.3.3 Visualization of key performance metrics

In Figures 6.7, 6.8, and 6.9 we propose a Campaign Performance Visualization (or Visualization of Key Performance Metrics) Module, which is included in order to present to the end user a more focused-monitoring of his campaigns.

[8] http://classifier4j.sourceforge.net/
[9] http://jgap.sourceforge.net/
[10] http://www.ee.ucl.ac.uk/~mflanaga/java/
[11] https://developers.google.com/AdWords/api/
[12] http://code.google.com/p/google-api-ads-java/

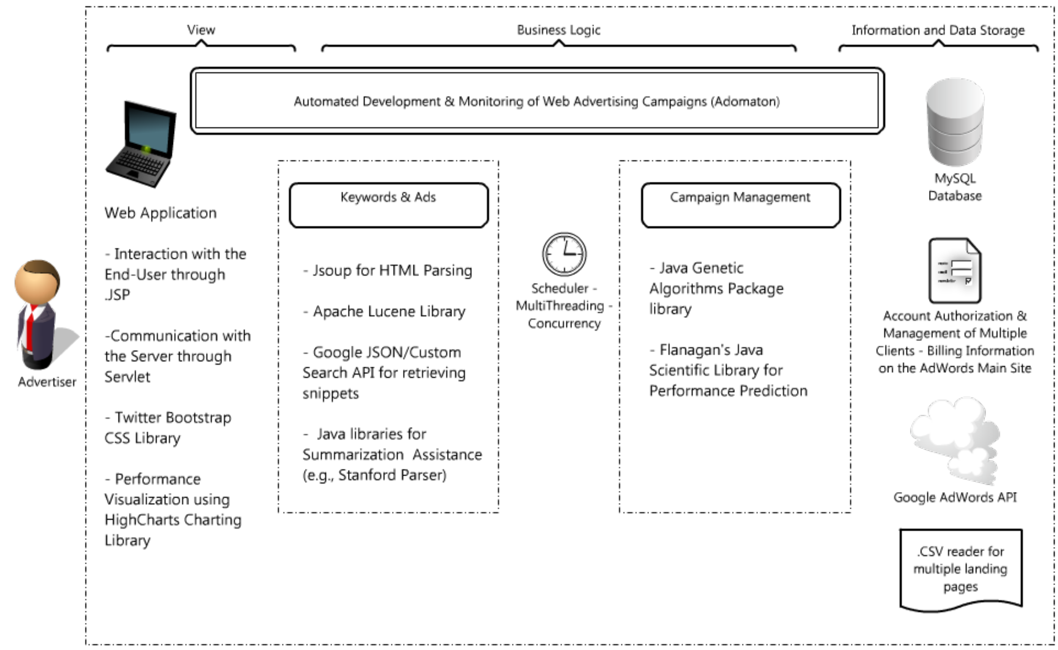


FIGURE 6.6: System Architecture

In this way, the advertisers can be notified with a direct indication of best performing AdGroup in terms of traffic or monetary profit, with the min-max bid representation per day which is helpful for the advertiser in order to understand and react quickly to the dynamics of a time-period (e.g., Greek car rental companies may want to attract more users during summer periods, bigger competition, higher bids), and finally with the cumulative ROI after the end of each Period. The latter is calculated based on ROI:

$$\frac{gainOfInvestment - costOfInvestment}{costOfInvestment} \tag{6.1}$$

Thus, for the Traffic ROI we have :

$$\frac{(clicks \times maxCPC) - (clicks \times avgCPC)}{clicks \times avgCPC} \tag{6.2}$$

and for the Monetary Profit ROI we have:

$$\frac{(conversions \times profitPerSale) - (clicks \times avgCPC)}{clicks \times avgCPC} \tag{6.3}$$

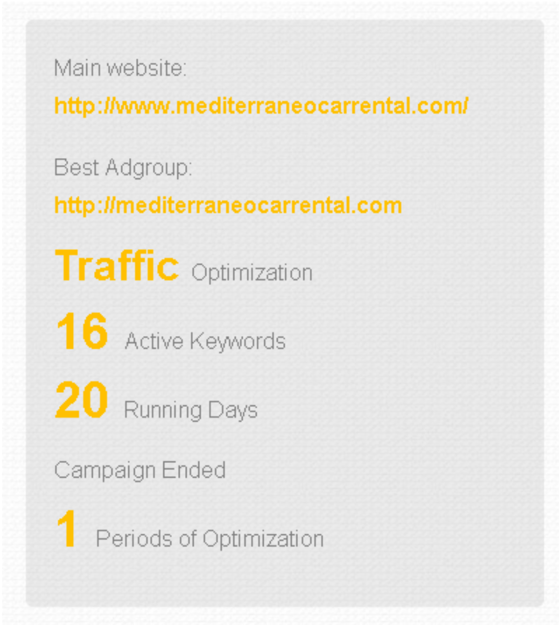


FIGURE 6.7: Focused Progress Statistics

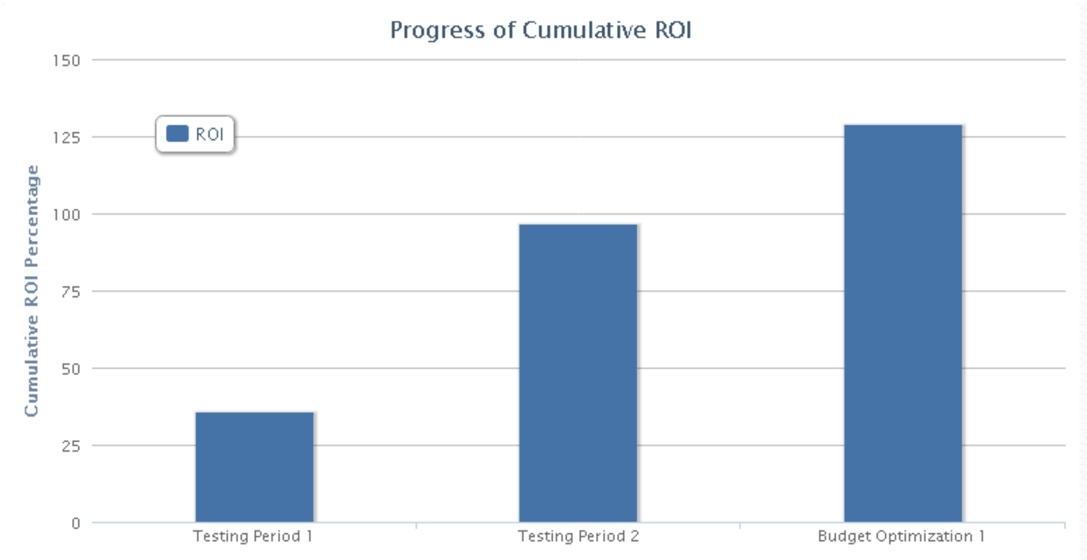


FIGURE 6.8: Cumulative ROI

6.3.4 User Authorization

Using OAuth 2.0 protocol[13], it is easier for every single user to trust and share his own data with our system. Our decision to use Google Client API comes with the fact that in order to manage someone's Google AdWords Campaigns, the end user has to be registered in this service, provided by Google. Consequently, the users are supposed to have a Google Account. Authenticating

[13] https://developers.google.com/adwords/api/docs/guides/authentication

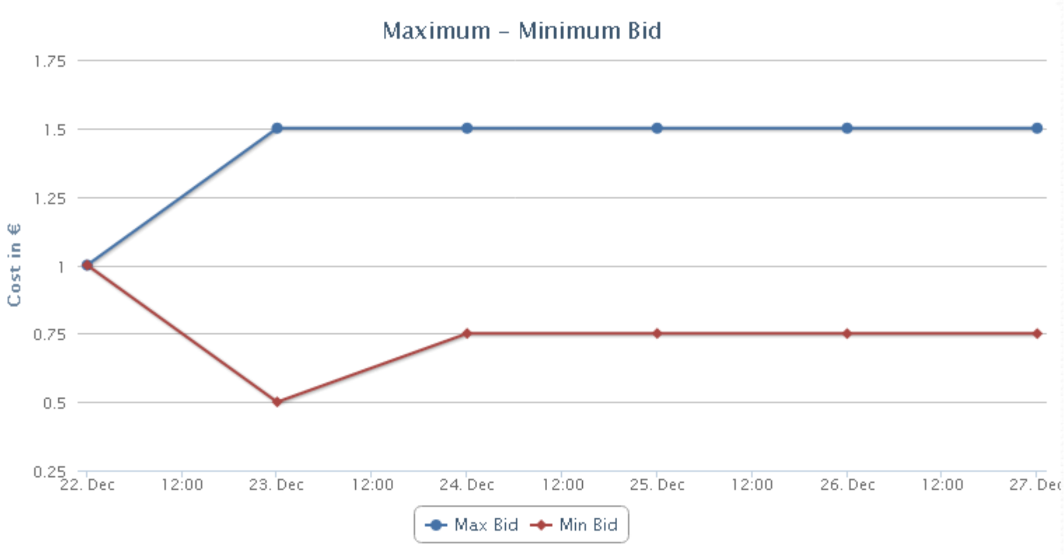


FIGURE 6.9: Min-Max bid per day

users with their Google Accounts, helps them to easier maintain credentials and trust the encryption of Google API due to the transfer of responsibility in the AdWords platform. In this way, it is much easier to use a single account credentials instead of having many accounts. Using their Google Accounts, users are also not obliged to register with a new account with another system. After users being authenticated, they can authorize us to use specified data, in order to be able to manage their Google AdWords Campaigns. When users try to login to our system for the first time, they will be redirected to authenticate their Google Account and authorize our web application system, letting us access specific data that they agree on. Afterwards, Google sends to our web platform an authorization code, which our system has to exchange in order to get the users' credentials. That way we can retrieve users' access and refresh tokens in a secure manner and more importantly retrieve Customer AdWords ID, which is obligatory for the campaign access and management, not only when the users are online, but also offline. This process is repeated once in a while in order to be sure that the logged in user is the one who uses the corresponding account.

6.3.5 AdWords API Dependency

Here we describe some critical issues that we came across during some phases of the development and the communication with the AdWords API. A first issue was that we had to migrate to new versions of AdWords API between our experiments in different phases due to the constant changes of this API every few months. The AdWords API needs a dedicated and constant notice and care for the API Migration.

Other factors that may interfere with certain changes on its service requests are related to the rate limiting: To ensure reliable access to the AdWords API, the system enforces a queries per second rate (QPS) that prevents software from maliciously or unintentionally overloading the servers. If an application exceeds this QPS limit (which varies based on server load and other variables), the server will return an error. A general good practice is to batch operations together into fewer requests. Making a request to the API has certain fixed costs, such as network transfer, serialization and deserialization, calls to backend systems, etc. Batching multiple operations into a single request lessens the impact of these fixed costs and increases overall performance. The mutate methods in the API are designed to accept an array of operations, so a good practice is to avoid making single-operation requests when possible. Take the example of adding 5000 keywords to a campaign, across multiple ad groups. Instead of making 5000 requests with 1 keywords each, make 10 requests with 500 keywords each. There are limits on the number of operations allowed in a request, and it may be needed to adjust the batch size to achieve optimal performance. An application such as our system that updates keyword-level bids can benefit from using sparse updates, as only the ad group ID, criterion ID, and bids field would need to be populated in the request. In a test using 150 keywords, a 20% performance increase was seen using sparse updates instead of passing the fully populated objects.

In order to surpass the structural issue of the potential same assignment of IDs in different campaign and adgroups that belong to another hierarchy account, we define and track those campaigns and adgroups through the IDs assigned by our database and not directly from the AdWords assigned values.

There are limitations concerning the Externalities table and the retrieved values from the AdWords API as well as their corresponding usage into the Linear Regression Prediction. Thus, a guideline is to introduce an additional checking if there are available to the AdWords API users the aforementioned values. If the Genetic Algorithm has select too few options, the prediction cannot be executed due to the shortage of historical knowledge of data (i.e., problem with degrees of freedom [42] for the Linear Regression). A future guideline in order to solve this issue is to take into consideration not only the statistical options produced by the Genetic Algorithm, but the total statistics.

6.4 System Evaluation

We conducted an overall evaluation of the Adomaton Prototype in order to test the performance of the system as a whole. We developed an automated campaign using the Adomaton interface for a Greek car rental company. A human administrator was responsible for a manual setup of a second campaign assisted by some baseline changes in the course of the experiment from the optimization tool of AdWords. The experiment was conducted for a period of 14 days[14].

6.4.1 Strategy Setup

The sequential process of the bidding strategy can be found mainly inside the SystemTasks class. The Genetic Algorithm can be fed only with options that have received clicks. So the KeywordOptions that are given as input to Genetic Algorithm must have received clicks. In order to give a chance and test also previously used keywords that have not received any clicks, we must retrieve from the database those keywords of which the summary of clicks inside the Statistics table sum up to zero and test them first through some initial testing periods before run an optimization task. The following is a guide through the model strategy as well as how we set up in our experiments the bidding values.

Experimental Scenario

1. *Perform Initial Testing*: Select 7 new keywords. Try them for testing period = 1 day. Run & Collect statistics.
2. *Second Initial Testing* (slightly modified than the previous step): Select 7 new keywords. Keep also the previous selected 7 keywords but modify 2 random previous keywords (update in AdWords their bids, so in

[14]Manual: November 2012 and Automated: December 2012. We could not have simultaneously two different accounts for the same website

the AdWords we have now 5 unedited and 2 update in bids + 7 new keywords = 14 keywords in the AdWords) with different bid. Try them for testing period = 1 day. Run & Collect Statistics.

- At this time, we must have 7+7+2 = 16 used keywords (be careful that are tested in different time periods, so they will have different start & end dates)

3. *Perform Budget Optimization*: Use some of the previous options for "feeding" the genetic algorithm (GA). The constraint here is to consider as candidate options only those options which have gained clicks during the previous testing periods. Let's assume that from the previous 16 options, the 12 have gained clicks. Thus, 12 options will be considered as input for the GA.

   - The genetic algorithm will select the most profitable solution set of options. Let's assume that the GA will produce a list of 7 options. These keywords must stay inside the used keywords table, as well as maintaining their status at Google AdWords. The other 12 – 7 = 5 keywords that were not luckily enough to be selected from the GA must be: a. deleted from the used keyword table, b. paused from AdWords. The rest 7 selected from GA keywords are the ones that are going to be tested now.
   - Regarding the remaining 4 options : We give them a chance, by letting them run along with the produced options of the GA. We define them as "extra".
   - Try all the above keywords for budget optimization period = 4 days. Run & Collect Statistics.
   - If there were no options to feed the GA, the GA would not run. In this case, ignore Step 3 and proceed to Step 4.

4. *Perform Testing After Optimization*: Select 3 new keywords. From the previous selected by the GA tested 7 keywords + 4 "extra" = 11 total keywords , leave 11-2=9 without editing them, and select random 2 in order to test them with different bids (update them also in AdWords). Thus, 3+11 = 14 keywords must be in the used keywords table. Try them for testing period = 1 day. Run & Collect statistics.

5. *Perform Budget Optimization*: Repeat from Step 3.

Bidding Values
Start with the default bid. The default bid in all cases where there cannot be defined or calculated properly a $maxEstimatedFirstPageCPC$ bid is 1.0 monetary unit (e.g., dollar or euro). During the random modification of the algorithm fluctuate by 50% its current bidding value.

6.4.2 Results

In Table 6.1 we present the overall comparison between performance metrics for the automated (Adomaton campaign) and the manual campaign. The high exposure of both campaigns in impressions results from the fact that we were experimenting in the field of *car rental*, one of the most competitive areas due to the high interest in queries from users. We observe that the Adomaton campaign outperforms overall the manual one.

Campaign	Clicks	Impressions	CTR	Avg. CPC
Adomaton	120	23960	0.50%	0.92
Manual	83	21449	0.39%	1.15

TABLE 6.1: Adomaton vs. Manual Campaign Performance

In Figure 6.10 we present the performance of the two campaigns in the course of time in terms of CTR. The Adomaton outperformed the manual campaign. This means that our selected keywords and ad creatives (both in the initial creation step as well as after the selection of the GA optimization) were constantly more relevant and attractive to the users in order to click to the advertisement. A remark here was that the manual campaign had a sharp performance drop and did not receive any clicks after the 8th day in contrast with the Adomaton one that continued to have good CTR values.

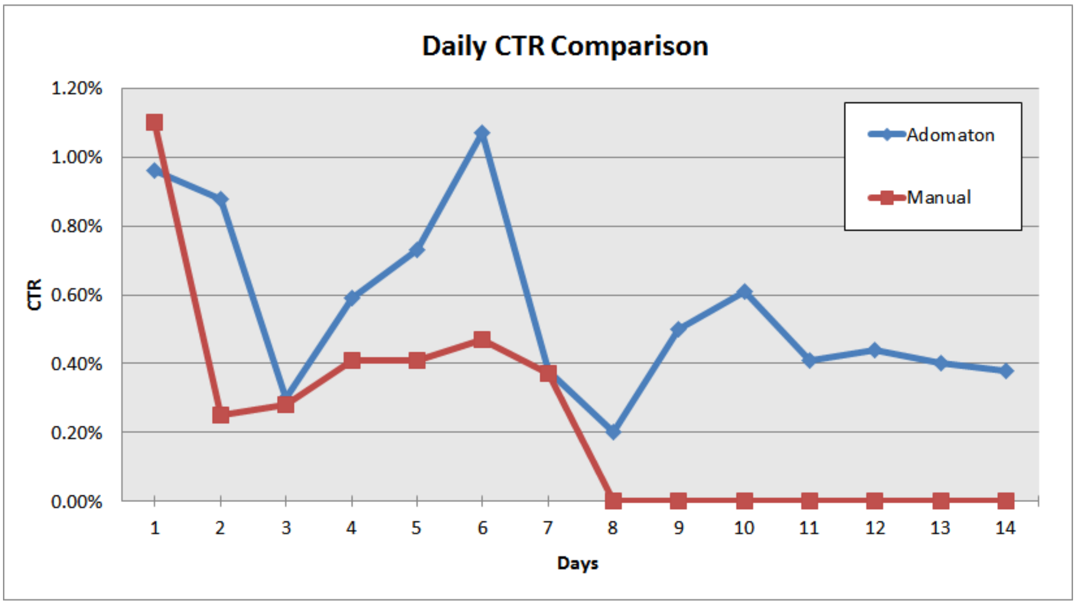


FIGURE 6.10: Daily CTR Comparison of Adomaton vs. Manual Campaign

6.5 Emerging Online Advertising Models

In addition to the research literature which was studied in the previous chapters, there exist also competitive systems and similar ideas to the modules of our system. Wordstream[15] and AdGooroo[16] determine an advertiser's top competitors and then actively search for the keywords they are targeting. After a period of time, lists of targeted keywords that are competitive for pay per click advertising are automatically generated. Special attention should be given to the fact these two approaches may result to a recommendation set of keywords which are likely to be general and thus more expensive. Criteo[17] enables online businesses to follow up visitors who have left their website without making a purchase using personalized banners which aim to drive potential customers back to the business website. AdGrok –acquired by Twitter[18]- was a tool that simplified the process of setting up Google AdWords campaigns. It's "Grokbar" was letting customers look at any page on their websites and see data about the Google AdWords campaigns that point to it — including the cost of each campaign and how well it is working.

[15] http://www.wordstream.com/
[16] http://www.adgooroo.com/
[17] http://www.criteo.com/
[18] http://mashable.com/2011/05/31/twitter-acquires-adgrok/

During the last couple of years new channels have appeared for providing advertising space and hence proposing new frameworks for online advertising campaign platforms. An emerging way of selling and buying ads on the Internet is via an exchange that brings sellers (publishers) and buyers (advertisers) together to a common, automatic marketplace [69] . There are exchanges in the world for trading financial securities to currency, physical goods, virtual credits, and much more. Exchanges serve many purposes from bringing efficiency, to eliciting prices, generating capital, aggregating information etc. Ad exchanges recent examples are RightMedia[19], adBrite[20], OpenX[21], and DoubleClick[22].

An *Ad Exchange* corresponds to the platform that facilitates automated auction based pricing and buying in real-time (e.g., Google, Facebook, Twitter). A *Demand-Side Platform (DSP)* corresponds to a system that allows digital advertisers to manage multiple ad exchange and data exchange accounts through one interface (e.g., Invite Media). A *Real Time Bidding* system corresponds to an agent software which consolidates access to multiple inventory sources and supports real-time bidding protocols (e.g., DoubleClick Bid Manager). This enables buyers to evaluate and bid on any available impressions in real-time. We can claim that the Adomaton prototype might play actually the role of a Real Time Bidder for ads on Google SERP.

[19] `http://www.rightmedia.com`
[20] `www.adbrite.com`
[21] `http://www.openx.com/`
[22] `http://www.google.com/doubleclick/`

Chapter 7

Conclusions

7.1 Summary

The main motivation for this thesis was to develop methods towards automating the full life cycle of online advertising campaigns and tackle each of the corresponding tasks. The principal goal was to propose an architecture and a prototype framework for automated advertising campaign development, monitoring, and optimization, putting emphasis on the campaign creation, management, and budget optimization modules. In summary, our contributions were the following.

The first aspect addressed in this thesis was the *Keyword Generation* task. We have proposed a system that, given only a landing page in the context of products and services promotion, extracts and suggests keywords for web advertising campaigns. In this task, our contribution regarding the improvement of advertising campaign developing process focused on:

- Automating the task of finding the appropriate keywords.
- Recommending multiword terms (bigrams, trigrams) with high specificity without the need to capitalize on usage data such as query and web traffic logs.
- A fully developed system with convincing experimentation on real world data from various thematic areas compared to the prominent competitive systems.

- Using the search result snippets for the process of keyword suggestion has helped a lot to retrieve faster the proper information rather than crawling actual documents. It was also a helpful mean to keep the trends and thus retrieving trending topics at a specific time.

The second aspect addressed was the *Ad-Text Generation* task, an open problem in the area of sponsored search advertising. Thus, the corresponding module of our system is an innovative contribution in this regard. We proposed an engine which produces compact ad-text snippets in an automated and massive manner given a product landing page as input. Such a system aims at facilitating the process of online advertising. The main notion was to provide an efficient solution for online marketing campaigns that feature large websites or shops that can be considered as large online product catalogues. These sites may include hundreds or thousands of products or services that each one of them need to be promoted through a text ad. At the same time, there is an emerging need for promotion through channels that require more and more short promotional text like interfaces on tablets and smartphones. In this way, our method contributes with the automated generation of compact but comprehensive ad text. In this task, we proposed the following:

- Compose combinations of n-grams after mining the most important phrases that can represent the promoted product or service.
- N-grams transformation to obtain well-formed candidates and thus construct an appropriate ad-text sentence.
- Find promising snippet candidates with Information and Readability Scoring, based also on a trigram language model trained on ads.
- Leverage sentiment analysis for keeping the most positive snippets that will have a good impact on the product image.

The third aspect addressed was the *Budget Optimization* task. We approximated the solution in this problem capitalizing on a genetic algorithm for budget optimization with multiple keyword options. We also proposed the use of keyword statistics to predict keyword behavior using multiple linear regression. Both a. the use of a genetic algorithm and b. impressions prediction for this type of problem form innovative solutions with respect to existing literature. The budget optimization problem, even though it is an NP-hard problem, has been approximately solved by modeling it as a multiple-choice knapsack problem. Another novelty in our study was that we focused on the advertisers and not explicitly on the other bidders or the self-interested auctioneer as the vast literature in this area does. Nevertheless, we gained an implicit knowledge from the auctioneer with two important variables: a. global monthly searches and b. competition for each campaign keyword. We used these parameters for observing and predicting the campaign behavior in favor of the advertiser.

Finally, proof of concept was given with the implementation of the proposed overall architecture. We developed a functional prototype system for Google AdWords platform campaigns, which currently occupies a vast share of websearch advertising volume. A comprehensive experimental evaluation was conducted on a simulated environment as well as on real world data. Our experimental results show that the automated campaigns from our engine overall outperform the manual competitive ones.

7.2 Future Work

Our proposed system in this thesis opens up new interesting issues of research in the areas of Text Mining, Information Retrieval, and Ad Auctions. The tools and techniques proposed as part of this platform can be developed and transformed further in order to serve new areas of online advertising and marketing.

7.2.1 Keyphrase Extraction and Creation of Ad Snippets

Both the tasks of *Keyword and Ad-Text Generation* can benefit from a named-entity recognition or an aspect term extraction task given the appropriate training data [79] in order to identify automatically more information about a product e.g., price, offers, or new features. Regarding the Sentiment Analysis task, it would be interesting to investigate a more sophisticated method for filtering out the negative candidates [66]. Regarding the marketing appeal of the snippet language it could be used two language models: One trained on background, more generic corpus in order to evaluate the general readability and one trained on a specific category topic of ads in order to evaluate features of more topical coherence. One interpolation method commonly used is Jelinek-Mercer smoothing [48] which considers each document to be a mixture of a document-specific model and a more general background model. Each document model is estimated using the maximum likelihood estimate of the terms in the document, linearly interpolated with a background language model. Another interesting approach could be experimenting with automatic text summarization techniques for summarizing the content of the given landing page and then generate paraphrases [7] from the resulted sentences to produce more ad-text candidates. Taking into consideration the aforementioned limitations of ad-texts lengths we could use sentence compression such as the method described in [35].

7.2.2 Budget Optimization - Campaign Performance

Other methods for solving the budget optimization task that can relate to our formulation of the problem would be to explore the potentials of a reinforcement learning method such as Contextual Bandit Learning [57] in order to exploit the various campaign features and test alternate and more deterministic bidding strategies in order to compare their performance, based on integer linear programming techniques

In order to predict the campaign behaviour and performance it would be interesting to test more methods and apply, for example, Expectation Maximization [107] techniques to cluster the keyword data, include more features in the bidding strategy such as location features [62], or apply a Hidden Markov Model [9] to see if there are any transitions in keyword state that could be predicted. Additionally, the prediction of clicks could be achieved using boosted regression trees [28, 32, 37] as more recent works are applying them for user click modeling and CTR prediction tasks [38, 100, 103].

7.2.3 Adomaton Prototype System Expansion

Our framework could be a basis for other engines related to advertising platforms and individual tasks. Individual modules could be used in other platforms e.g., ad snippets for Facebook or promoted tweets through the usage of

RESTful web services or a dedicated library. Other external software systems could be using our algorithms through API calls. For example, the *BudgetOptimization* methods could return a set of options like <*keywords: state(active or paused), bids*>. Finally, it would be interesting to study the optimization of an existing campaign, by importing statistics and overall structure.

List of Figures

List of Tables

Abbreviations

API	**A**pplication **P**rogramming **I**nterface
CPA	**C**ost **P**er **A**cquisition
CPC	**C**ost **P**er **C**lick
CPM	**C**ost **P**er **M**ille
CR	**C**onversion **R**ate
CTR	**C**lick **T**rough **R**ate
DF	**D**ocument **F**requency
GA	**G**enetic **A**lgorithm
GSP	**G**eneralized **S**econd **P**rice
IDF	**I**nverse **D**ocument **F**requency
JSON	**J**avaScript **O**bject **N**otation
JSP	**J**ava **S**erver **P**age
LM	**L**anguage **M**odel
LP	**L**inear **P**rogramming
PMI	**P**ointwise **M**utual **I**nformation
PPA	**P**ay **P**er **A**cquisition
PPC	**P**ay **P**er **C**lick
PPM	**P**ay **P**er **M**ille
QPS	**Q**ueries **P**er **S**econd
QS	**Q**uality **S**core
REST	**R**epresentational **S**tate **T**ransfer
ROI	**R**eturn **O**f **I**nvestment
SEM	**S**earch **E**ngine **M**arketing
SEO	**S**earch **E**ngine **O**ptimization

SERP	**S**earch **E**ngine **R**esult **P**age
SOAP	**S**imple **O**bject **A**ccess **P**rotocol
TF	**T**erm **F**requency

Appendix A

Adomaton Database Schema Details

This appendix provides a table overview of the Adomaton database schema. In Tables A.1 - A.10 we provide details for the table fields and what they represent.

A description of the entities is the following:

Account represents a discrete entity of a user. The Adomaton system in order to have access to the data of each user account must be authorized first from the user through the OAuth 2.0 Authentication Protocol.

Adgroup represents an AdWords set of keywords, ads, and bids that is a key part of how the campaign is organized. Each ad campaign is made up of one or more ad groups.

Adtext represents the standard type of AdWords ad. A text ad typically includes a link to your website and a description or promotion of your product or service.

Campaign represents a set of general preferences and total budget for the advertising purpose.

Externalities represent provided values by AdWords that are beneficial for capturing external factors and conditions of the ad auctions and are being used in the Impressions Prediction in order to predict current or future behavior.

Keyword represents the AdWords keyword: A word or phrase that matches a web-user's search query and at the same time describes the advertised content. This entity is responsible for holding the keyword matching option, which is the mechanism that controls which searches can trigger the ad. The definition of each match type, in order from broad to narrow is the following:

1. *Broad*: The default matching option. The ad may show if a search term contains the keyword terms in any order, and possibly along with other terms. The ads can also show for close variations of the keywords.
2. *Phrase*: The ad can show when someone searches for the exact keyword, or the exact keyword with additional words before or after it. The ad can show also when someone searches for close variations of that exact keyword, or with additional words before or after it.
3. *Exact*: The ads can appear only when someone searches for the exact keyword, without any other terms in the search. The ad can show when someone searches for close variations of that specific keyword.

Relevance corresponds to a keyword recommendation from our system.

Statistics represent statistical information collected regarding the performance of the keywords.

Task represents the functionalities that must be executed from the Scheduler.

UsedKeyword represents a selected keyword by our bidding strategy that remained active during a defined period of time along with its tested bid on this period.

Field	Description
accountId	The ID of the account for the database
customerAdWordsId	The corresponding value ID from the AdWords
accessToken	Access Token from the OAuth 2.0
refreshToken	Refresh Token from the OAuth 2.0
userName	
userEmail	

TABLE A.1: Account Entity

Field	Description
adgroupId	The ID of the AdGroup for the database
adgroupAdWordsId	The corresponding value ID from the AdWords
campaignId	It correlates with the Campaign Database ID
suburl	A specific landing page
profitPerSale	A monetary value for the profit that will be gained through a conversion on this very specific landing page

TABLE A.2: Adgroup Entity

Field	Description
adTextId	The ID of the AdText for the database
adTextAdWordsId	The corresponding value ID from the AdWords
adgroupId	It correlates with the AdGroup Database ID
headline	The problem or opportunity; Ad titles are limited to 25 characters
description1	Short description of big benefit; limited to 35 characters
description2	Short description of the product/service; limited to 35 characters
displayUrl	The web site's name up to 35 characters; Google can only display up to 35 characters of the display URL, due to limited space. If the display URL is longer than 35 characters, it will appear shortened when the ad is displayed
actualUrl	Landing page

TABLE A.3: Adtext Entity

Field	Description
campaignId	The ID of the Campaign for the database
campaignAdWordsId	The corresponding value ID from the AdWords
accountId	It correlates with the Account Database ID
url	Main Website just for reference purposes
period	Total active period
startDate	Starting Date
endDate	Ending Date
budget	Total Budget
advertisingGoal	Use Cases of the System: 1. No Optimization 2. Traffic Optimization 3. Profit Optimization
advertisingTarget	Marketing Target helpful only for the call-to-action phrases of the ad creative: 1. Website/Brand-name 2. Product 3. Service

TABLE A.4: Campaign Entity

Field	Description
externalityId	The ID of the Externality for the database
text	Keyword text
competition	Will of competition of all the other bidders upon this specific keyword. Takes values in the range [0,1]
targetedMonthlySearches	Previous called by AdWords Global Monthly Searches (GMS): Average user searches for this keyword query
retrievedDate	Retrieved date of the statistic

TABLE A.5: Externalities Entity

Field	Description
keywordId	The ID of the Keyword for the database
keywordAdWordsId	The corresponding value ID from the AdWords. Notice: It is like a hashed-code value. E.g. the keyword "car rental" has the same AdWords id for all the accounts, campaigns, adgroup that are using it
relevanceId	It correlates with the relevance Database ID
adgroupId	It correlates with the AdGroup Database ID
text	Keyword text
matchType	Keyword Matching Option: Control which searches can trigger the ad: 1. BROAD 2. PHRASE 3. EXACT

TABLE A.6: Keyword Entity

Field	Description
relevanceId	The ID of the Keyword Recommendation for the database
adgroupId	It correlates with the AdGroup Database ID
text	The recommended text
relevance	Relevance score. Takes values in the range [0,1]
initialBid	Initial Bid that is recommended when the keyword is been generated
tested	A Boolean value [0 or 1] if it has been tested on AdWords

TABLE A.7: Relevance Entity

Field	Description
statisticsId	The ID of the Statistic for the database
keywordId	It correlates with the Keyword Database ID
adgroupId	It correlates with the AdGroup Database ID
text	Keyword Text
startDate	Starting date of the retrieved statistic
endDate	Ending date of the retrieved statistic
maxCPC	
averageCPC	
impressions	
clicks	
conversions	
cost	

TABLE A.8: Statistics Entity

Field	Description
taskId	The ID of the Task for the database
campaignId	It correlates with the Campaign Database ID
type	A code number for the task type 1. InitialTestingOfKeywordsId 2. PerformBudgetOptimizationId 3. TestingAfterOptimizationId
startMillis	Real starting time (milliseconds from Epoch)

TABLE A.9: Task Entity

Field	Description
usedKeywordId	The ID of the Used Keyword for the database
keywordId	It correlates with the Keyword Database ID
adgroupId	It correlates with the AdGroup Database ID
text	Keyword Text
testedBid	Tested bid
startDate	Starting date of this testing combination (keyword, bid)
endDate	Ending date of this testing combination (keyword, bid)

TABLE A.10: UsedKeyword Entity

Bibliography

[1] V. Abhishek and K. Hosanagar. Keyword generation for search engine advertising using semantic similarity between terms. In *Proceedings of the Ninth International Conference on Electronic Commerce*, ICEC '07, pages 89–94, 2007.

[2] V. Abhishek and K. Hosanagar. Optimal bidding in multi-item multislot sponsored search auctions. *Operations Research*, 61(4):855–873, 2013.

[3] Z. Abrams and M. Schwarz. Ad auction design and user experience. In *Proceedings of the 3rd International Conference on Internet and Network Economics*, WINE'07, pages 529–534, 2007.

[4] G. Aggarwal, A. Goel, and R. Motwani. Truthful auctions for pricing search keywords. In *Proceedings of the 7th ACM Conference on Electronic Commerce*, EC '06, pages 1–7, 2006.

[5] A. Anagnostopoulos, A. Z. Broder, E. Gabrilovich, V. Josifovski, and L. Riedel. Just-in-time contextual advertising. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, CIKM '07, 2007.

[6] A. Anagnostopoulos, A. Z. Broder, E. Gabrilovich, V. Josifovski, and L. Riedel. Web page summarization for just-in-time contextual advertising. *ACM TIST*, 3(1):14, 2011.

[7] I. Androutsopoulos and P. Malakasiotis. A survey of paraphrasing and textual entailment methods. *J. Artif. Intell. Res. (JAIR)*, 38:135–187, 2010.

[8] K. Bartz, C. Barr, and A. Aijaz. Natural language generation for sponsored-search advertisements. In *Proceedings of the 9th ACM Conference on Electronic Commerce*, EC '08, pages 1–9, 2008.

[9] J. Bilmes. A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Technical report, 1998.

[10] C. Borgs, J. Chayes, N. Immorlica, K. Jain, O. Etesami, and M. Mahdian. Dynamics of bid optimization in online advertisement auctions. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 531–540, 2007.

[11] L. Bottou, C. Cortes, J. Denker, H. Drucker, I. Guyon, L. Jackel, Y. LeCun, U. Muller, E. Sackinger, P. Simard, and V. Vapnik. Comparison of classifier methods: a case study in handwritten digit recognition. In *Proceedings of the 12th IAPR International Conference on Computer Vision amp; Image Processing*, volume 2, pages 77–82 vol.2, 1994.

[12] A. Broder, M. Fontoura, V. Josifovski, and L. Riedel. A semantic approach to contextual advertising. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 559–566. ACM, 2007.

[13] A. Broder, E. Gabrilovich, and V. Josifovski. Information retrieval challenges in computational advertising. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, 2010.

[14] A. Z. Broder, P. Ciccolo, M. Fontoura, E. Gabrilovich, V. Josifovski, and L. Riedel. Search advertising using web relevance feedback. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, pages 1013–1022, 2008.

[15] J. Carletta. Assessing agreement on classification tasks: The kappa statistic. *Comput. Linguist.*, 22(2):249–254, June 1996.

[16] S. Chen, D. Beeferman, and R. Rosenfeld. Evaluation Metrics for Language Models. In *DARPA Broadcast News Transcription and Understanding Workshop (BNTUW)*, 1998.

[17] S. F. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4):359–393, 1999.

[18] Y. Chen, G.-R. Xue, and Y. Yu. Advertising keyword suggestion based on concept hierarchy. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, WSDM '08, pages 251–260, 2008.

[19] Y. Choi, M. Fontoura, E. Gabrilovich, V. Josifovski, M. Mediano, and B. Pang. Using landing pages for sponsored search ad selection. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 251–260, 2010.

[20] K. W. Church and P. Hanks. Word association norms, mutual information, and lexicography. In *Proceedings of the 27th Annual Meeting on Association for Computational Linguistics*, ACL '89, pages 76–83, 1989.

[21] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292, 2001.

[22] O. Damani. Improving pointwise mutual information (pmi) by incorporating significant co-occurrence. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 20–28, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

[23] B. Di Eugenio and M. Glass. The kappa statistic: A second look. *Comput. Linguist.*, 30(1):95–101, Mar. 2004.

[24] W. Ding, T. Qin, X.-D. Zhang, and T.-Y. Liu. Multi-armed bandit with budget constraint and variable costs. In *AAAI*, 2013.

[25] F. Djannaty and S. Doostdar. A hybrid genetic algorithm for the multidimensional knapsack problem.

[26] A. Douzet. An Investigation of Pay Per Click Search Engine Advertising: Modeling the PPC Paradigm to lower Cost per Action. *cdn.theladders.net*, pages 1–7.

[27] B. Edelman, M. Ostrovsky, and M. Schwarz. Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. *American Economic Review*, 97(1):242–259, March 2007.

[28] J. Elith, J. R. Leathwick, and T. Hastie. A working guide to boosted regression trees. *Journal of Animal Ecology*, 77(4):802–813, 2008.

[29] E. Even Dar, V. S. Mirrokni, S. Muthukrishnan, Y. Mansour, and U. Nadav. Bid optimization for broad match ad auctions. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 231–240, 2009.

[30] J. Feldman, S. Muthukrishnan, M. Pal, and C. Stein. Budget optimization in search-based advertising auctions. In *Proceedings of the 8th ACM Conference on Electronic Commerce*, EC '07, pages 40–49, 2007.

[31] K. Fjell. Online advertising: Pay-per-view versus pay-per-click with market power. *Journal of Revenue and Pricing Management*, 9:198–203, 2010.

[32] J. H. Friedman. Stochastic gradient boosting. *Comput. Stat. Data Anal.*, 38 (4):367–378, 2002.

[33] A. Fujita, K. Ikushima, S. Sato, R. Kamite, K. Ishiyama, and O. Tamachi. Automatic generation of listing ads by reusing promotional texts. In *Proceedings of the 12th International Conference on Electronic Commerce: Roadmap for the Future of Electronic Business*, ICEC '10, pages 179–188, 2010.

[34] E. Gabrilovich. Ad retrieval systems in vitro and in vivo: Knowledge-based approaches to computational advertising. In *Proceedings of the 33rd European Conference on Advances in Information Retrieval*, ECIR'11, pages 4–5, 2011.

[35] D. Galanis and I. Androutsopoulos. An extractive supervised two-stage method for sentence compression. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 885–893, 2010.

[36] K. Ganesan, C. Zhai, and E. Viegas. Micropinion generation: An unsupervised approach to generating ultra-concise summaries of opinions. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 869–878, 2012.

[37] Y. Ganjisaffar, R. Caruana, and C. V. Lopes. Bagging gradient-boosted trees for high precision, low variance ranking models. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 85–94, 2011.

[38] Y. Gao, J. Pan, G. Ji, and F. Gao. A time-series modeling method based on the boosting gradient-descent theory. *Science China Technological Sciences*, 54(5):1325–1337, 2011.

[39] N. Gatti, A. Lazaric, and F. Trovò. A truthful learning mechanism for contextual multi-slot sponsored search auctions with externalities. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, EC '12, pages 605–622, 2012.

[40] B. Geddes. *Advanced Google AdWords*. SYBEX Inc., Alameda, CA, USA, 1st edition, 2010.

[41] I. Gluhovsky. Forecasting click-through rates based on sponsored search advertiser bids and intermediate variable regression. *ACM Trans. Internet Technol.*, 10(3):11:1–11:28, 2010.

[42] I. J. Good. What are degrees of freedom? *The American Statistician*, 27(5): 227–228, Dec. 1973.

[43] V. Gopalakrishnan, S. P. Iyengar, A. Madaan, R. Rastogi, and S. H. Sengamedu. Matching product titles using web-based enrichment. In *CIKM*, pages 605–614, 2012.

[44] Y.-C. J. H.-F. Yu, C.-H. Ho and C.-J. Lin. *LibShortText: A Library for Short-text Classification and Analysis*, 2013.

[45] D. Hillard, S. Petersen, and M. Ostendorf. N-gram language modeling tutorial. Lecture Notes.

[46] X. Hu, N. Sun, C. Zhang, and T.-S. Chua. Exploiting internal and external semantics for the clustering of short texts using world knowledge. In *CIKM*, pages 919–928, 2009.

[47] K. Hui, B. Gao, B. He, and T.-j. Luo. Sponsored search ad selection by keyword structure analysis. In *Proceedings of the 35th European Conference on Advances in Information Retrieval*, ECIR'13, pages 230–241, 2013.

[48] F. Jelinek and R. L. Mercer. Interpolated estimation of markov source parameters from sparse data. In *Proceedings of the Workshop on Pattern Recognition in Practice*, pages 381–397, Amsterdam, The Netherlands: North-Holland, May 1980.

[49] P. R. Jordan and M. P. Wellman. Designing an ad auctions game for the trading agent competition. In *Agent-Mediated Electronic Commerce. Designing Trading Strategies and Mechanisms for Electronic Markets*, volume 59 of *Lecture Notes in Business Information Processing*, pages 147–162. 2010.

[50] A. Joshi and R. Motwani. Keyword generation for search engine advertising. In *Proceedings of the Sixth IEEE International Conference on Data Mining - Workshops*, ICDMW '06, pages 490–496, 2006.

[51] D. Jurafsky and J. H. Martin. *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Prentice Hall, 2000.

[52] D. Jurafsky and J. H. Martin. *Speech and language processing, 2nd edition*. Prentice Hall, 2008.

[53] S. Kiritchenko and M. Jiline. Keyword optimization in sponsored search via feature selection. In *FSDM*, volume 4 of *JMLR Proceedings*, pages 122–134, 2008.

[54] C. Lawson and R. Hanson. *Solving least squares problems*. Classics in applied mathematics. 1995.

[55] K. Lee, D. Palsetia, R. Narayanan, M. M. A. Patwary, A. Agrawal, and A. Choudhary. Twitter trending topic classification. In *Proceedings of the 2011 IEEE 11th International Conference on Data Mining Workshops*, ICDMW '11, pages 251–258, 2011.

[56] H. Li and J. D. Leckenby. Internet advertising formats and effectiveness. 2004.

[57] L. Li, W. Chu, J. Langford, and R. E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 661–670, 2010.

[58] K. Liakopoulos, S. Thomaidou, and M. Vazirgiannis. The adomaton prototype: Automated online advertising campaign monitoring and optimization. In *Eighth Ad Auctions Workshop, 13th ACM Conference on Electronic Commerce*, 2012.

[59] J. Lin, R. Snow, and W. Morgan. Smoothing techniques for adaptive online language models: topic tracking in tweet streams. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '11, pages 422–429, 2011.

[60] J. Liu, C. Wang, Z. Liu, and W. Yao. Advertising keywords extraction from web pages. In *Proceedings of the 2010 international conference on Web information systems and mining*, WISM'10, pages 336–343, 2010.

[61] A. Louis and T. Newman. Summarization of business-related tweets: A concept-based approach. In *COLING (Posters)*, pages 765–774, 2012.

[62] D. Lymberopoulos, P. Zhao, C. Konig, K. Berberich, and J. Liu. Location-aware click prediction in mobile local search. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, pages 413–422, 2011.

[63] Z. Ma, A. Sun, Q. Yuan, and G. Cong. Topic-driven reader comments summarization. In *CIKM*, pages 265–274, 2012.

[64] M. Mahdian and K. Tomak. Pay-per-action model for on-line advertising. *Int. J. Electron. Commerce*, 13(2):113–128, Dec. 2008.

[65] P. Maillé, E. Markakis, M. Naldi, G. D. Stamoulis, and B. Tuffin. Sponsored search auctions: An overview of research with emphasis on game theoretic aspects. 12(3):265–300, Sept. 2012.

[66] P. Malakasiotis, R. M. Karampatsis, K. Makrynioti, and J. Pavlopoulos. nlp.cs.aueb.gr: Two stage sentiment analysis. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 562–567, Atlanta, Georgia, USA, June 2013. Association for Computational Linguistics.

[67] C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA, 1999.

[68] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. 2008.

[69] Y. Mansour, S. Muthukrishnan, and N. Nisan. Doubleclick ad exchange auction. *CoRR*, abs/1204.0535, 2012.

[70] P. Marshall and B. Todd. *Ultimate Guide to Google AdWords 3/E: How to Access 100 Million People in 10 Minutes*. McGraw-Hill Education, 2012.

[71] S. Martello and P. Toth. *Knapsack problems: algorithms and computer implementations*. John Wiley & Sons, Inc., New York, NY, USA, 1990.

[72] Y. Matsuo and M. Ishizuka. Keyword extraction from a single document using word co-occurrence statistical information. *International Journal on Artificial Intelligence Tools*, 13:2004, 2004.

[73] A. Mehta, A. Saberi, U. Vazirani, and V. Vazirani. Adwords and generalized online matching. *J. ACM*, 54(5), 2007.

[74] M. Mitchell. *An introduction to genetic algorithms*. MIT Press, Cambridge, MA, USA, 1996.

[75] D. C. Montgomery, E. A. Peck, and G. G. Vining. *Introduction to Linear Regression Analysis, Solutions Manual (Wiley Series in Probability and Statistics)*. 2007.

[76] M. Munsey, J. Veilleux, S. Bikkani, A. Teredesai, and M. D. Cock. Born to trade: A genetically evolved keyword bidder for sponsored search. In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2010, Barcelona, Spain, 18-23 July 2010*, pages 1–8, 2010.

[77] S. Muthukrishnan. Adx: A model for ad exchanges. *SIGecom Exch.*, 8(2): 9:1–9:2, Dec. 2009.

[78] S. Muthukrishnan, M. Pál, and Z. Svitkina. Stochastic models for budget optimization in search-based advertising. pages 131–142, 2007.

[79] J. Pavlopoulos and I. Androutsopoulos. Aspect term extraction for sentiment analysis: New datasets, new evaluation measures and an improved unsupervised method. In *Proceedings of the 5th Workshop on Language Analysis for Social Media (LASM 2014), 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2014)*, pages 44–52, 2014.

[80] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st Annual International ACM SIGIR*

Conference on Research and Development in Information Retrieval, SIGIR '98, pages 275–281, New York, NY, USA, 1998. ACM.

[81] A. Rajaraman and J. D. Ullman. *Mining of Massive Datasets*. Cambridge University Press, New York, NY, USA, 2011.

[82] A. Rangrej, S. Kulkarni, and A. V. Tendulkar. Comparative study of clustering techniques for short text documents. In *WWW (Companion Volume)*, 2011.

[83] S. Ravi, A. Broder, E. Gabrilovich, V. Josifovski, S. Pandey, and B. Pang. Automatic generation of bid phrases for online advertising. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, WSDM '10, pages 341–350, 2010.

[84] S. E. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at trec-3. In *TREC*, 1994.

[85] M. Rocha and J. Neves. Preventing premature convergence to local optima in genetic algorithms via random offspring generation. In *Multiple Approaches to Intelligent Systems*, volume 1611 of *Lecture Notes in Computer Science*, pages 127–136. 1999.

[86] P. Rusmevichientong and D. P. Williamson. An adaptive algorithm for selecting profitable keywords for search-based advertising services. In *Proceedings of the 7th ACM Conference on Electronic Commerce*, EC '06, pages 260–269, 2006.

[87] L. Sarmento, P. Trezentos, J. a. P. Gonçalves, and E. Oliveira. Inferring local synonyms for improving keyword suggestion in an on-line advertisement system. In *Proceedings of the Third International Workshop on Data Mining and Audience Intelligence for Advertising*, ADKDD '09, pages 37–45, 2009.

[88] R. Schimmel. Keyword suggestion for search engine marketing.

[89] D. Shen, J.-D. Ruvini, and B. Sarwar. Large-scale item categorization for e-commerce. In *CIKM*, pages 595–604, 2012.

[90] Y. Song, H. Wang, Z. Wang, H. Li, and W. Chen. Short text conceptualization using a probabilistic knowledgebase. In *IJCAI*, 2011.

[91] A. Sood, T. P. Mohamed, and V. Varma. Topic-focused summarization of chat conversations. In *ECIR*, pages 800–803, 2013.

[92] R. Stokes. *Mastering Search Advertising: How the Top 3Advertisers Dominate Google AdWords*. 2008.

[93] A. Stolcke. SRILM – an extensible language modeling toolkit. In *Proceedings of ICSLP*, volume 2, pages 901–904, Denver, USA, 2002.

[94] B. Szymanski. Impact of ROI on bidding and revenue in sponsored search advertisement auctions. In *Workshop on Sponsored Search Auctions*, pages 1–8, 2006.

[95] S. Thomaidou and M. Vazirgiannis. Multiword keyword recommendation system for online advertising. In *Proceedings of the 2011 International Conference on Advances in Social Networks Analysis and Mining*, ASONAM '11, pages 423–427, 2011.

[96] S. Thomaidou, K. Leymonis, K. Liakopoulos, and M. Vazirgiannis. Admad: Automated development and optimization of online advertising campaigns. In *Proceedings of the 2012 IEEE 12th International Conference on Data Mining Workshops*, ICDMW '12, pages 902–905, 2012.

[97] S. Thomaidou, K. Leymonis, and M. Vazirgiannis. Grammads: Keyword and ad creative generator for online advertising campaigns. In *Digital Enterprise Design and Management 2013*, volume 205 of *Advances in Intelligent Systems and Computing*, pages 33–44. Springer Berlin Heidelberg, 2013.

[98] S. Thomaidou, I. Lourentzou, P. Katsivelis-Perakis, and M. Vazirgiannis. Automated snippet generation for online advertising. In *Proceedings of the*

22nd ACM international conference on Conference on information & knowledge management, CIKM '13, pages 1841–1844, 2013.

[99] S. Thomaidou, K. Liakopoulos, and M. Vazirgiannis. Toward an integrated framework for automated development and optimization of online advertising campaigns. *Intell. Data Anal.*, 18(6):1199–1227, Nov. 2014.

[100] S. Tyree, K. Q. Weinberger, K. Agrawal, and J. Paykin. Parallel boosted regression trees for web search ranking. In *Proceedings of the 20th International Conference on World Wide Web, WWW 2011, Hyderabad, India, March 28 - April 1, 2011*, WWW.

[101] T. Weise. Global optimization algorithms theory and application, 2008.

[102] I. H. Witten, G. W. Paynter, E. Frank, C. Gutwin, and C. G. Nevill-Manning. Kea: Practical automatic keyphrase extraction. In *Proceedings of the Fourth ACM Conference on Digital Libraries*, DL '99, pages 254–255, 1999.

[103] K.-W. Wu, C.-S. Ferng, C.-H. Ho, A.-C. Liang, C.-H. Huang, W.-Y. Shen, J.-Y. Jiang, M.-H. Yang, T.-W. Lin, C.-P. Lee, P.-H. Kung, C.-E. Wang, T.-W. Ku, C.-Y. Ho, Y.-S. Tai, I.-K. Chen, W.-L. Huang, C.-P. Chou, T.-J. Lin, H.-J. Yang, Y.-K. Wang, C.-T. Li, S.-D. Lin, and H.-T. Lin. A two-stage ensemble of diverse models for advertisement ranking in kdd cup 2012. 2012.

[104] X. Yan, J. Guo, Y. Lan, and X. Cheng. A biterm topic model for short texts. In *WWW*, pages 1445–1456, 2013.

[105] W.-t. Yih, J. Goodman, and V. R. Carvalho. Finding advertising keywords on web pages. In *Proceedings of the 15th International Conference on World Wide Web*, WWW '06, pages 213–222, 2006.

[106] S. Yuan, A. Z. Abidin, M. Sloan, and J. Wang. Internet advertising: An interplay among advertisers, online publishers, ad exchanges and web users. *CoRR*, abs/1206.1754, 2012.

[107] P. Zacharouli, M. K. Titsias, and M. Vazirgiannis. Web page rank prediction with pca and em clustering. In *Algorithms and Models for the Web-Graph, 6th International Workshop, WAW 2009, Barcelona, Spain*, pages 104–115, 2009.

[108] H. Zaragoza, N. Craswell, M. J. Taylor, S. Saria, and S. E. Robertson. Microsoft cambridge at trec 13: Web and hard tracks. In *Proceedings of the Thirteenth Text REtrieval Conference, TREC 2004, Gaithersburg, Maryland, November 16-19, 2004*, 2004.

[109] Y. Zhang, N. Zincir-Heywood, and E. Milios. World wide web site summarization. *Web Intelli. and Agent Sys.*, 2(1):39–53, Jan. 2004.

[110] N. Zhou, J. Wu, and S. Zhang. A keyword extraction based model for web advertisement. In *Integration and Innovation Orient to E-Society Volume 2*, volume 252 of *IFIP Advances in Information and Communication Technology*, pages 168–175. Springer Boston, 2007.

[111] Q. Zhou and W. Luo. A novel multi-population genetic algorithm for multiple-choice multidimensional knapsack problems. In *Proceedings of the 5th International Conference on Advances in Computation and Intelligence*, ISICA'10, pages 148–157, 2010.

[112] Y. Zhou and V. Naroditskiy. Algorithm for stochastic multiple-choice knapsack problem and application to keywords bidding. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pages 1175–1176, 2008.

[113] Y. Zhou, D. Chakrabarty, and R. Lukose. Budget constrained bidding in keyword auctions and online knapsack problems. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pages 1243–1244, 2008.