MACHINE LEARNING-BASED EXTRACTION OF KEYPHRASES AND ITS APPLICATIONS IN MULTIPLE DOMAINS

A DISSERTATION
SUBMITTED TO THE PHD SCHOOL IN COMPUTER SCIENCE
OF UNIVERSITY OF SZEGED
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Superviors: Prof. János Csirik , Dr. Richárd Farkas

Gábor Berend
June 2014

Preface

Raw data of any form conveys no information unless it is processed in some intelligent way. Knowing the most important phrases of textual documents can provide a condensed representation of them which can considerably ease their processing. However, the manual determination of the sets of important phrases for every single document in a large collection of documents is a tedious and expensive task and it often requires expert knowledge. **Natural language processing** techniques – mostly relying on **machine learning** – can fortunately help the automatic generation of keyphrases for documents.

In this thesis, various models for the extraction of keyphrases from textual documents of various genres and languages are presented, and their potential end-application utilization is demonstrated in the form of a document visualization system. Although most of the earlier studies focused on the domain of scientific papers, we will introduce models for the extraction of keyphrases in two languages (i.e. English and Hungarian) and from various genres including scientific publications, news articles and product reviews as well.

Acknowledgments

First of all, I would like to thank my supervisors, János Csirik and Richárd Farkas, for their guidance and for supporting my work with useful comments.

I am indebted to my senior colleagues who showed me interesting undiscovered fields and helped give birth to new ideas during our inspiring discussions. In alphabetical order: Márk Jelasity, Róbert Ormándi, György Szarvas and Veronika Vincze.

I would also like to thank my colleagues and friends who helped me to realize the results presented here and to enjoy my period of PhD study at the University of Szeged. In alphabetical order: István Hegedűs, György Móra, István Nagy.

I would also like to thank David P. Curley for scrutinizing and correcting this thesis from a linguistic point of view.

I would like to thank my girlfriend Kata for her endless love, support and inspiration. Last, but not least, I wish to thank my parents and my sister for their constant love and support. I would like to dedicate this thesis to them as a way of expressing my gratitude and appreciation.

This work was supported by the European Union and the State of Hungary, co-financed by the European Social Fund in the framework of TÁMOP 4.2.4. A/2-11-1-2012-0001 'National Excellence Program'. I am grateful for this support, which definitely acted as an accelerator for the submission of this thesis.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

"Essentially, all models are wrong, but some are useful."
George E. P. Box

The pace at which data is generated nowadays – and often made available for on-line access to almost everyone – has encouraged the introduction and application of intelligent and automated ways to process data. According to the report [42] published in 2012 on the predicted size and the future growth of the digital universe, it has most likely exceeded 3 Zetabytes by now and it will reach 40 Zb (i.e. 40 billion terabytes) in 2020. This expectation nicely illustrates the rapid growth of data constantly being produced.

Data items today are generated by an extremely wide audience and range – including geolocational, acceleration measurement data – mostly due to smart devices. All the diverse and immense data that is continuously being produced is often referred as *big data* nowadays. This new scale of data not only called for the creation of novel, distributed techniques with respect their storage (see Hadoop Distributed File System [100] for example), but in their processing as well (see MapReduce [30]). An interesting expectation owing to the Yahoo! spin-off (called Hortonworks) is that more than half of all the world's data will be processed by Apache Hadoop by the end of 2015[1].

Despite the abundant appearance of novel types of data, it is still extensively produced in the good-old textual format. These textual data items may take many forms, ranging from (micro)blog and forum posts to news portal entries and scientific literature. Documents originating from the different genres can differ greatly – in their length, writing style, degree of structure in them, and so on. However, what they all have in common is that large quantities of them are accessible and that their detailed processing is practically impossible without machine-augmented methods.

As the amount of digitally available textual data continues to grow exponentially, the need for

[1] https://twitter.com/williammcknight/status/251001949804167170

automated techniques to process and retrieve information from them becomes increasingly important. One effective way of representing the main contents of documents is by describing them in the form of keyphrases. In this thesis, various models for the extraction of keyphrases from textual documents of various genres and languages are presented and also their end-application utilization is demonstrated.

The characteristic of keyphrases for describing and summarizing the contents of documents in a compressed form makes them very appealing for several natural language processing tasks. They can be extremely useful in the categorization, summarization and retrieval of textual documents. Analyzing the relative importance of keyphrases over time offers the possibility of performing trend detection, and the aggregation of document-level keyphrases to provide keyphrases for multiple documents can be utilized in intelligent visualization tools, as will be demonstrated later on.

Despite their potential utility, most of the documents are not supplied with keyphrases and their manual assignment to documents is time-consuming and costly, hence methods for their automated generation are desirable. For this reason, the extraction of keyphrases from documents has gained academic interest recently.

Although most of the previous studies have focused on the domain of scientific papers, this thesis will introduce models for the extraction of keyphrases in two languages (i.e. English and Hungarian) and from various genres including those of news articles and product reviews.

1.1 Keyphrase generation

Variants of the task of automatic keyphrase generation can be formalized in multiple ways. Generally speaking, we would like to find a function k which determines a set of useful keyphrases K_i to document d_i, i.e. $k(d_i) = K_i$. We can think of set D_i which contains all the possible subsequences of tokens, i.e. n-grams of arbitrary lengths, that are present in document d_i.

Let the set C_i consist of the candidate phrases (e.g. n-grams retrievable from a document up to a certain length) belonging to document d_i. In practice, the relation $C_i \subseteq D_i$ always holds. Furthermore, let K_i^* be the set of gold standard keyphrases of document d_i. This set can be obtained from various sources, i.e. gold standard keyphrases might be regarded as those which were assigned to a document by its authors or by some of its readers (see e.g. [55]). Gold standard keyphrases – although possibly being less reliable – might even be derived from social tagging sites, such as `CiteULike.org`, as was done in [79].

As a final notation during the formal discussion of keyphrase generating techniques, let I be a set of index terms, the members of which are regarded *a priori* as phrases with the possibly to act as keyphrases on some document domain (e.g. scientific articles taken from the field of *game theory*). In the absence of any prior knowledge about the possible keyphrases, we can simply define a non-informative set of index terms by defining $I = \bigcup_{j \in \mathbb{N}} \Sigma^j$, i.e. the infinite set consisting of all the

possible character sequences of the alphabet Σ.

Imposing certain conditions on K_i – being the set of keyphrases returned for document d_i by mapping k – different approaches of automatic keyphrase generation can be distinguished:

- **Keyphrase assignment**: In this setting $K_i \subseteq \bigcup_{j \neq i} K_j^*$, meaning that the keyphrases assigned to a document are such that they are known to be gold standard keyphrases with respect some other document. Note that this approach does not require keyphrases returned for a document to be actually present in it, i.e. even $D_i \cap K_i = \emptyset$ might hold. Approaches of this kind might be referred as **keyphrase recommendation** systems.

- **Keyphrase indexing**: In this setting $K_i \subseteq C_i \cap I$, meaning that the keyphrases proposed for document d_i should be present in it and be a member of some predefined list of index terms as well.

- **Keyphrase extraction**: In this setting $K_i \subseteq C_i$, the only difference being to keyphrase indexing is that here the existence of some predefined list of index terms is not assumed (or equivalently, a non-informative, infinite list of index terms is assumed).

After formally defining keyphrase generation paradigms, we will briefly describe them. We should add that keyphrase indexing and keyphrase extraction differ only in the informativeness of the set of index terms, hence they will be discussed together.

1.1.1 Keyphrase assignment

Keyphrase assignment or *tag recommendation* systems including AutoTag [81] and TagAssist [101] rely mainly on previously tagged corpora. The key idea behind these approaches is that upon assigning tags to documents, the tags of the most similar documents are applied. AutoTag, the pioneering work of tag recommendation, applies standard information retrieval metrics to find similar documents and chooses tags from the most similar ones based on frequency information. Participants of past ECML PKDD tag recommendation challenges also built their systems on document-similarity-based approaches (see [36, 107]).

Such methods, however, have the disadvantage of exploiting tags assigned by humans that are often inappropriate (e.g. users often assign ineffective tags to articles such as "*to read*") or inconsistent with other parts of the document set due to their highly specialized nature. Moreover, these approaches cannot really adapt to the dynamics of topics, as they cannot introduce new tags. This is because they operate on a predefined set of tags that were previously assigned to at least one document. Another drawback of these methods is that they are heavily domain dependent, meaning that every time we wish to use them on some new document set, vast amounts of labeled documents from the same genre are required.

1.1.2 Keyphrase indexing and extraction

As mentioned in the formal definition of *keyphrase indexing* and *keyphrase extraction* methods, the former require the generated keyphrases to be present on an informative list of possible keyphrases, while the latter does not impose such a requirement on the extracted keyphrases. The KEA++ framework [78] is the keyphrase indexing variant of the keyphrase extraction system KEA [119], meaning that the keyphrases returned for a document not only need to be present in the document, but have to be included in a domain-dependent thesaurus as well. The use of a thesaurus might prevent ill-formed phrases from being handled as keyphrases, but it can also rule out otherwise correct keyphrases from the returned set of phrases simply due to the incompleteness of the thesaurus. Also, thesauri for some topics are not necessarily easy to access.

As the existence of domain-dependent thesauri cannot be guaranteed and their generation can be expensive, throughout this thesis we will not rely on them and only focus on keyphrase extraction tasks instead.

1.1.3 Evaluation

As noted in [125], the evaluation of keyphrase generating systems most often occurs by

1. relying on the *manual evaluation* of keyphrases performed by human judges, or
2. applying some *in vivo* (i.e. application-oriented) evaluation, or
3. performing automated, *in vitro* experiments on the set of predicted keyphrases.

During a manual evaluation, human judges have to decide which of the predicted keyphrases are suitable for describing the main contents of a document (i.e. which ones would they accept as a keyphrase for a particular document). Although this kind of evaluation is expected to accurately reflect the real-world utility of the predicted keyphrases, it can rarely be employed due to the tedious and expensive nature of the work required.

When application-oriented evaluation is employed, we measure the usefulness of a keyphrase-extraction module by evaluating a complex application, which relies on the automatic keyphrase extraction system, the performance of which we are interested in. The performance of different keyphrase extraction submodules can be seen by evaluating entire pipelines using different submodules for generating keyphrases. The drawback of an evaluation like this is that the performance scores do not simply reflect the quality of the submodule we are interested in, but rather the entire pipeline of applications. Also, keyphrases which help some end-application to achieve better scores are not necessarily those that human judges would always find better.

Automatic evaluation seeks to qualify the predicted keyphrases by comparing them to some gold standard reference set of keyphrases of the test documents. Gold standard keyphrases generally originate from two sources: they might be those determined by the author(s) of the documents or

they might be assigned to the document by its independent readers. The different sources of gold standard keyphrases are often referred to as *author,- and reader-defined gold standard* keyphrases. *Combined gold standard* sets are also employed, which is simply the union of the two kinds of sets.

The quality of K_i, being the keyphrases determined by some automated method for document d_i, is characterized by comparing it with the corresponding set of gold standard keyphrases, K_i^*. Based on the interaction of these sets for each document i, we can define the values

- *true positive*: $TP_i = |K_i \cap K_i^*|$,
- *false positive*: $FP_i = |K_i \setminus K_i^*|$ and
- *false negative*: $FN_i = |K_i^* \setminus K_i|$.

These scores can then be combined to get the document-level precision(P_i) and recall(R_i) scores, i.e.

$$P_i = \frac{TP_i}{TP_i + FP_i}$$

and

$$R_i = \frac{TP_i}{TP_i + FN_i}.$$

Assuming that a test set consists of N documents, the overall precision and recall scores can be defined by either relying on the aggregation of document-level TP_i, FP_i and FN_i values or by averaging the document-level P_i and R_i scores. The former strategy is referred to as *micro-averaging*, while the latter is called *macro-averaging*, which can be formulated in the following way:

$$P_{micro} = \frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N} TP_i + FP_i}, \qquad P_{macro} = \frac{1}{N} \sum_{i=1}^{N} P_i$$

$$R_{micro} = \frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N} TP_i + FN_i}, \quad R_{macro} = \frac{1}{N} \sum_{i=1}^{N} R_i.$$

Next, test set-level precision and recall values are combined into an F-score by calculating the harmonic mean of the two values, i.e. $F = \frac{2*P*R}{P+R}$. Depending on which precision and recall values are used during the calculation of the previous ratio, *micro,-or macro-F-scores* can be calculated. Unless stated otherwise, throughout this thesis we will follow the practice of reporting F-scores in their micro-averaged form.

As mentioned above, manual evaluation is not a real alternative for the evaluation of keyphrase extraction due to the amount of human effort required. Furthermore, as end-application-based evaluation can be biased by factors other than the quality of the predicted keyphrases, we chose automatic evaluation as our primary evaluation technique. But we should mention that in Chapter 5, we also report results based on human evaluation (performed on the subsets of our test corpora), and the results in Chapter 6 – focusing on the possible applications of keyphrase extraction – can be regarded as an end-application evaluation of our keyphrase extraction models.

Issues with automatic evaluation

As automatic evaluation is our main evaluation criterion, we shall discuss some of its characteristics, including some of its shortcomings. Here, we list the potential issues for the automatic evaluation of keyphrase extraction systems and we shall provide domain-specific examples to illustrate these issues later on.

Automatic evaluation – relying on simple set operations over K_i and K_i^* – has weak capabilities of handling the morphological and lexical variability of phrases having the same meaning. One can easily think of a situation where an automated system treats a phrase as a keyphrase that is not among the set of gold standard keyphrases, but in fact is in synonymy or hyponymy/hypernymy relations with some element of the set of gold standard keyphrases. In such a situation, automatic evaluation relying on strict evaluation (i.e. exact string matching) would account for one false positive and one false negative, instead of counting for one true positive, if semantic equivalences were taken into account (e.g. by human inspection).

Calculating the intersection and difference between sets K_i and K_i^* based on approximate string matching (like that done in [25, 125]) can weaken this characteristic of automatic evaluation. However, by doing so we would risk regarding predicted phrases as true positives even in such cases where they should be regarded as false positives. To avoid this, we demanded that the string representation of a predicted keyphrase match exactly a gold standard keyphrase in order to be accepted as a true positive.

The strong requirement on exact string matching makes the evaluation prone to severely underestimating the potential real-world usefulness of predicted keyphrases as the false negative rate would most likely be lower if human evaluation was employed.

Despite its tendency to underestimate the quality of keyphrases, scores calculated by automatic evaluation tend to strongly correlate with those of human evaluation. This suggests that even though the scores obtained during automatic evaluation are expected to be lower than what a human evaluation would produce, the relative differences among the scores of different keyphrase extraction systems should resemble those of a human evaluation.

1.2 Structure of the dissertation

In Chapter 2, we provide an introduction to machine learning and its connection with natural language processing. The remainder of the dissertation is comprised of two main parts. The first part deals with the generation of keyphrases from textual documents of various genres and languages and the second part illustrates how the outputs of these models can be utilized in applications.

In Chapter 3, we introduce the problem of retrieving keyphrases from documents originating from news articles, a genre being more heterogeneous from both topical and stylistic perspectives compared to scientific publications – which is the most common domain for performing keyphrase

			Chapter			
Venue	Year		3	4	5	6
SEMEVAL	2010	[8]	•			
RANLP	2011	[84]	•			
NLE	2014	2014	•			
ECAI	2010	[38]		•		
IJCNLP	2011	[6]			•	
RANLP	2011	[12]			•	
WASSA	2012	[11]			•	
CICLing	2013	[10]				•
IJCNLP	2013	[9]				•

Table 1.1: The relation between the thesis topics and the corresponding publications

extraction. Another special point of this chapter is that it describes the extraction of keyphrases from documents written in Hungarian.

In Chapter 4, we describe a machine learning-based feature rich keyphrase extraction framework for the standard setting of keyphrase extraction from English scientific documents. The novel features presented in this part of the thesis are designed to capture the semantic orientation of keyphrases by means of linguistic analysis and external knowledge sources as well.

In Chapter 5, we introduce the task of extracting keyphrases from yet another quite dissimilar genre, namely opinionated utterances from product review sites. Here, we validate our assumption that *pro* and *con* phrases assigned to product reviews function similar to keyphrases of non-opinionated documents. Furthermore, we propose useful extensions to general keyphrase extraction models to achieve better performance measures on this specialized, opinion mining-related keyphrase extraction task.

In Chapter 6, we focus on the utilization of the outputs of the keyphrase extraction models described in the previous chapters. The possible applications described here are a technique for assigning sets of keyphrases to document subcorpora (in contrast to single documents) and a keyphrase-based corpus visualization approach.

In Table 1.1, we list the author's key publications related to this thesis. For a full list of the author's publications, please visit `http://www.inf.u-szeged.hu/~berendg/?pp=publ`.

Chapter 2

Machine Learning and Natural Language Processing

In this chapter, we would like to discuss some of the machine learning techniques and concepts that are related to this thesis, and explain their connection with tasks of natural language processing, one special subtask of which is keyphrase extraction.

Machine learning is a very broad term that refers to the study of systems that can learn from data. Machine learning can be basically separated into two types, namely **supervised learning** and **unsupervised learning**. The whole area of machine learning has an extensive literature [15, 33, 47, 82, 83], but here, we will focus on those essential parts of it that relate to this thesis. In the following, the two main branches of machine learning and its subfields will be introduced briefly, focusing on techniques that will be applied later on.

2.1 Supervised machine learning

In **supervised learning** the goal is to learn a mapping from input vector $\mathbf{x}$ to output y. Supervised approaches use a **training set** of the form $\mathcal{D} = \{(\mathbf{x_i}, y_i)\}_{i=1}^{N}$, which is a set of labeled feature vector-training label pairs of cardinality N.

In the common notation $\mathbf{x_i}$ denotes the i^{th} training sample, which is a vector of **feature values** – mostly characterized by (real) numbers – representing that particular datum object. Here, we will assume that the training set consist of N instances, each instance being described by m features (not counting the target variable). We shall note that *feature values* are sometimes also referred as **descriptive** or **attribute values**.

y_i corresponds to the **response variable** that encodes the expected answer when seeing a given datum $\mathbf{x_i}$. In principle, there is no restriction on what y_i might be; it can take (real) value, or some

discrete (i.e. nominal or categorical) value (often encoded by numbers as well). The former case of predicting continuous values is called **regression** within supervised learning, whereas predicting discrete outcomes is a matter of **classification**. *Predicting housing prices* on a continuum scale can be thought of as a typical problem of regression, whereas *deciding on the credibility of a loan applicant* can be typically thought of as a classification task. The *response value*, reflecting the state of nature of the classification instances, is sometimes also referred to as the **output variable** or **target variable**.

Classification can be further divided into special cases depending on the total number of possible outcomes: we can distinguish **binary classification** – when the task is to decide whether or not an instance belongs to some particular category – from **multi-class** or **multinomial classification** when there are three or more classes an instance can belong to. Sometimes the *states of nature*, i.e. the class labels of a classification instance are not mutually exclusive, in which case y_is are most easily interpreted as sets instead of simple categorical values. The latter case is called **multi-label classification**, not to be confused with multi-class classification. Note that the state of nature assigned to an observation can be of any complexity: one can image lists or graphs that are treated as target observations.

Existing solutions for keyphrase extraction most often treat it as a binary classification task, where the aim is the decide whether a candidate phrase extracted from a document belongs to the class of proper or an improper keyphrases.

2.1.1 Maximum Entropy Modeling

Maximum Entropy (ME) modeling can be used for the task of classification, and there exist several descriptions of it from a natural language processing perspective. Among others, articles [13, 52] provide a thorough discussion of this approach.

Maximum entropy modeling can be regarded as the **discriminative** counterpart of the **generative** Naïve Bayes classification framework. The two techniques differ conceptually, as the parameters of the generative model – unlike those for discriminative models – are estimated in such a way as to maximize the (log)likelihood of the entire training dataset, i.e. that of the observation of both the indicator and target features at the same time. As this approach seeks to maximize the feature realizations and class labels jointly, generative models are often referred to as **joint classifiers**.

Discriminative classifiers, however, have a different objective, namely to maximize the *conditional* (log)likelihood of the observation of the class labels on the training dataset conditioned on the corresponding feature vectors, i.e. $\log \prod_{i=1}^{N} p(y_i|\mathbf{x_i})$. This is why discriminative models are also called **conditional models**.

The conceptual differences are depicted in Figure 2.1 and Figure 2.2 in the form of graphical models which include model dependency relations among descriptive and target variables in the form of a Bayes' net. As is common with Bayes' nets, nodes represent random variables (the shaded ones

(a) Generative graphical model (b) Discriminative graphical model

Figure 2.1: Bayes networks of the generative Naïve Bayes and the discriminative Maximum Entropy classifiers

(a) Generative graphical model (b) Discriminative graphical model

Figure 2.2: Plate notations of the generative Naïve Bayes and the discriminative Maximum Entropy classifiers

being observable, the unshaded ones being latent or directly unobservable) and directed edges among nodes denote probabilistic dependencies among the variables. Figure 2.2 differs from Figure 2.1 in that it uses the plate notation, which is a more compact representation, since the numbers of the variables in the plates get a multiplicative factor, as indicated on the plates. For more complex graphical models, this kind of notation is often more convenient than the verbose one.

The basic idea behind Maximum Entropy modeling is to find $p^*(y|\mathbf{x})$ out of the possible distributions of the class labels conditioned over the observable variables that has the highest conditional entropy, yet matches our empirical expectations of the feature counts based on the training data. These criteria together say that the aim is to find a model which is consistent with our training data with respect the feature count observations, but still includes as much uncertainty as possible via the maximization of conditional entropy, which has the form

$$H(y|x) = -\sum_{(\mathbf{x},y)\in\mathcal{X}\times\mathcal{Y}} p(\mathbf{x},y)\log p(y|\mathbf{x}) = \mathbb{E}_{p(\mathbf{x},y)}[-\log p(y|\mathbf{x})].$$

These desiderata can be incorporated into a constrained optimization problem that can be solved using Lagrange multipliers. The corresponding Lagrange function is then

$$\Lambda(p(y|\mathbf{x}),\vec{\lambda}) = H(y|\mathbf{x}) + \sum_{i=1}^{m}\lambda_i(E(f_i) - \hat{E}(f_i)) + \lambda_{i+1}\left(\sum_{y\in\mathcal{Y}} p(y|\mathbf{x}) - 1\right),$$

where $E(f_i)$ and $\hat{E}(f_i)$ correspond to the expected value and the empirical distribution of the i^{th} feature function out of m features. The first m constraints (i.e. one for each feature) simply state that the empirical and expected feature counts of the proposed model should match (i.e. it should be consistent with the training data) and the last constraint requires that the proposed conditional distribution be a valid distribution that sums to one.

Calculating the partial derivative of the Lagrange function yields

$$\frac{\partial \Lambda(p(y|\mathbf{x}), \vec{\lambda})}{\partial p(y|\mathbf{x})} = -\hat{p}(\mathbf{x})(1 + \log p(y|\mathbf{x})) + \sum_{i=1}^{m} \lambda_i \hat{p}(\mathbf{x}) f_i(\mathbf{x}, y) + \lambda_{m+1}, \tag{2.1}$$

where $\hat{p}(\mathbf{x})$ denotes the empirical feature distribution determined based on some training data.

The report in [58] contains a detailed derivation of how equating Equation 2.1 to zero implies the model formulation

$$p^*(y|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\Big(\sum_{i=1}^{m} \lambda_i f_i(\mathbf{x}, y)\Big),$$

$Z(\mathbf{x})$ being called the partition function, which ensures that we define a real probability distribution so then if we sum over $y \in \mathcal{Y}$ we get 1. This partition function has the form

$$Z(\mathbf{x}) = \sum_{y \in \mathcal{Y}} \exp\Big(\sum_{i=1}^{m} \lambda_i f_i(\mathbf{x}, y)\Big).$$

Note that Maximum Entropy modeling can be also thought of as an extension of regression for the task of classification. An alternative derivation from this perspective leads to a log-linear model, for which reason it is sometimes referred to as **logistic regression**. In Chapter 6 of [52], there is a derivation of the same model formulation from this perspective. This derivation assumes that the *log odds* of an instance belonging to some class is a linear function of the feature vector with some weight vector $\vec{\lambda}$. That is,

$$\log \frac{p(y = 1|\mathbf{x})}{1 - p(y = 1|\mathbf{x})} = \vec{\lambda}^{\intercal} \vec{x}.$$

Note that algorithms which generalize the idea of maximum entropy classification for structured objects (e.g. linear sequences) may be more suitable for structured prediction tasks. Such algorithms include conditional random fields (CRF) [60] and Maximum Entropy Markov Models (MEMM). One example task where these algorithms could be preferable is Part-Of-Speech (POS) tagging, where one intends to determine the POS-tags of the individual tokens of some natural language sentence.

Regularization

If we were satisfied with being able to reconstruct the target values of the training data there would be no need for machine learning at all as dictionary-lookup-based solutions would be able to achieve 100% performance on previously seen data (if we disregard memory consumption-related problems

that might arise when employing such an approach). However, as the main objective of inductive learning is to build models that are capable of predicting the target value for *unseen data*, special machinery needs to be employed.

The **bias-variance trade-off** is the reason why models that are otherwise highly accurate on some training data could perform poorly on test data. Due to this trade-off, one can increase classification accuracy on training data at the expense of increasing the model complexity. Doing so, however, contains the risk that our model would become highly specific towards the training data and prevent its capability of generalizing well. Unfortunately, classification performance on previously unseen test instances degrades severely if the model has a low capability of generalization. The phenomenon of finding a model that is highly specific towards the training data – at the expense of sacrificing generalization capability – is called **overfitting**.

The overall goal is thus to find the kind of models that are able to generalize patterns from training samples and also work accurately when they have to classify some previously unseen *test data*. This technique involves an extra component in the objective function (besides accounting for the cost of misclassified instances) to be optimized, namely the complexity of the model. This idea employs the **minimum description length** (MDL) criterion, meaning that "smaller" or less complex hypotheses should be preferred over the "larger" hypotheses. A way of avoiding problems that could arise from the incompleteness of the training sample is to employ **regularization** or **smoothing**, which penalizes hypotheses based on their extents of complexity.

With the help of regularization we are able to incorporate into the model our prior beliefs about the most likely values of the λ_i model parameters. These prior beliefs often assert that the parameter values have a *zero mean*. Parameter estimation which takes into account not only the data we have access to, but also the credibility of the model parameters with respect to our prior beliefs is called **maximum a posteriori** (MAP) parameter estimation. In contrast, parameter estimation that ignores prior beliefs and relies only on the observed training data is called **maximum likelihood** (ML) estimation. We should mention here that in the theoretical limit, i.e. when one has access to an infinite number of training samples, ML and MAP estimations produce the same results.

The above-mentioned notions are usually accompanied by terms "priors" and "MAP estimation" in the Bayesian language, whereas terms like (L_2 norm-based) "regularization" better suit the viewpoint of frequentist statisticians. Besides the regularization based on L_2 norm, different regularization techniques exist, including the L_1 norm-based LASSO technique. Due to the characteristics of the different distances applied during regularization, different effects can be produced. While regularization on the basis of L_1 norm favors sparse models, models regularized via the L_2-norm favor a "general" shortness of the weight vector, regardless of the proportion of 0 components in it.

For instance, upon the determination of the optimal weight vector of a logistic regression classifier which relies on the L_2-regularized *negative log likelihood function* of the training data, optimization

is of the form

$$\min_{\lambda} \sum_{i=1}^{N} \log\left(1 + \exp(-y_i \lambda^{\intercal} \mathbf{x_i})\right) + \alpha \parallel \lambda \parallel_2^2 .$$

Here, the parameter α controls the amount of regularization. Setting $\alpha = 0$ yields the original unregularized logistic regression objective function, while increasing the value of α affects how severely "large" models get penalized during the optimization.

Parameter estimation of the maximum entropy models discussed in this thesis was performed using the MALLET framework [77], employing L_1 regularization. Using L_1 regularization essentially neutralized the presence of less useful and potentially redundant features in the representation of keyphrase candidates.

Conditional sequence models were applied during the automatic linguistic analysis of textual documents when carrying out POS tagging and Named Entity recognition. The MEMM tagger [110] included in `Stanford CoreNLP` was used during the preprocessing of English texts. Stanford CoreNLP was also used to perform other linguistic analysis tasks, i.e. tokenization, lemmatization and syntactic parsing [57].

2.2 Unsupervised machine learning

The second main branch of machine learning is called **descriptive** or **unsupervised learning**. Here, no target variables are given during training time; instead input data is only given in the form of $\mathcal{D} = \{\mathbf{x_i}\}_{i=1}^{N}$, and the goal is to find useful and interesting patterns in the data. This is sometimes called **knowledge discovery** and algorithms used in this area are often called undirected (as y_is cannot guide the algorithms to find patterns). The present classification of unsupervised learning tasks follows that in [83]. This thesis relates to unsupervised learning via Chapter 6, as it employs clustering and the detection of latent factors, techniques introduced below.

2.2.1 Clustering

Clustering is the process of grouping data points based on their similarities. For example, we might want to group words that share some common semantic or syntactic behavior, as it was done by Brown et al. [22].

Clustering approaches can be further characterized as **hard** or **soft clustering** techniques. The basic difference between the two is that hard clustering algorithms assign each data point to exactly one group of points, whereas soft clustering allows data to be assigned to more than one clusters simultaneously. Latent Dirichlet Allocation (LDA) [17] for instance can be interpreted as a soft clustering technique, as it treats documents and words as distributions over topics they might belong to.

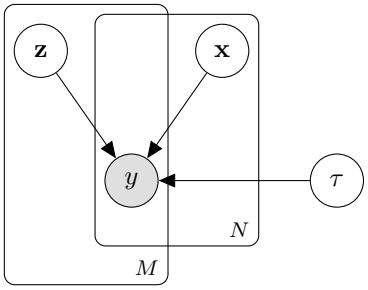


Figure 2.3: Plate notation of principal component analysis

When performing clustering in the absence of labeled data, the general framework of **Expectation Maximization** [31] is often employed. It makes it possible to perform maximum likelihood parameter estimation even if we only have access to incomplete data. Its general machinery can be related to the well-known **Jensen's inequality**, which ensures that the incomplete (log)likelihood of the input data never decreases during the iterative estimation of the model parameters.

2.2.2 Determining latent factors

Discovering latent factors can be helpful when data instances have a high dimensionality. Determining latent factors gives us the benefit of representing data in some advantageous manner in a lower dimensional subspace. Depending on how we define these advantageous representations, we will have various algorithms, including principal component analysis (PCA), independent component analysis (ICA), singular value decomposition (SVD) and categorical correlation analysis (CCA).

One well-known algorithm for determining latent factors is PCA. Figure 2.3 represents PCA as a graphical model with its plate diagram. From the model formulation we can see that the value of y – being an observable variable – is assumed to be dependent on three independent factors: the N-dimensional variable, $\mathbf{x}$, some general noise τ (possibly deriving from measurement) and most importantly, some hidden M-dimensional ($M << N$) "cause" z, which influences the data y. The task here is to determine this M-dimensional explanation of the data.

The main goal of PCA is to represent the original data in some lower dimensional subspace in such a way that the variance of the data is preserved as much as possible. The optimal projection of the data can be found by solving a constrained optimization problem. Doing so, we get that the projection should be performed by the transformation determined by the matrix which consists of the first k eigenvectors (corresponding to the largest eigenvalues) of the scatter (or covariance) matrix of the data. In Chapter 6 (where we demonstrate the possible applications of keyphrase extraction), we also relied on this technique.

2.3 Special machine learning tasks

Building a supervised model for classification tasks in the absence of abundant labeled training data will most likely result in a classifier which performs poorly on unseen test data. When the amount of labeled data is insufficient for supervised learning, the simplest solution is to collect additional labeled data. However, this approach is often unsuitable, as the collection of training data can be very expensive and time consuming.

Alternative solutions exist if we wish to avoid the tedious task of additional labeling. **Semi-supervised learning** and **domain adaptation** are two such alternatives that are discussed next.

2.3.1 Semi-supervised learning

The general goal of **semi-supervised learning** is the same as that of supervised learning, i.e. to build a model capable of predicting some target variable of unseen test data based on their predictive variables. The main difference is, however, that while the true values of the target variables of every training datum is known during supervised learning (disregarding some possible noise of the labels), semi-supervised algorithms have access only to $N' < N$ (usually $N' \ll N$) target values out of the total N training examples. Semi-supervised learning can thus be interpreted as a combination of supervised and unsupervised learning, meaning that the target values of the training instances are only partially observable.

2.3.2 Domain adaptation

Another way of overcoming the possible shortage of labeled training instances is to rely on **domain adaptation** techniques. Suppose we have a small amount of labeled training data for a specific task (*target domain*), but ew have a vast amount of labeled training data for another – not exactly the same, but somehow analogous – task (*source domain*). The idea here is that the performance on the target domain can be improved by transferring and incorporating knowledge from the source domain where the absence of labeled data does not cause a problem.

Imagine that we want to build a model that is capable of deciding whether a patient is suffering from a recently discovered (thus ill-documented) disease. Due to the ill-documented nature of the disease, we can assume that ample amount of training examples – for building some reliably performing model – is not available. However, if we have access to vast amounts of patient records suffering from some different, yet similar disease, we can use these samples to build our model for predicting whether someone is suffering from the former, poorly documented disease.

To put it more formally, in the case of domain adaptation, we are given two sets of instances, $\mathcal{D}_S \in \mathbb{R}^{m_S}$ and $\mathcal{D}_T \in \mathbb{R}^{m_T}$ during the training phase, $\mathcal{D}_S$ and $\mathcal{D}_T$ being data of the source and target domains, respectively. Note that data belonging to the different domains are not necessarily represented by the same sets of features. Furthermore, $|\mathcal{D}_S| \gg |\mathcal{D}_T|$ also typically holds for the

sizes of the two distinct domains. Under these circumstances, the task is then to transfer some valuable knowledge from the source domain in order to improve classification performance on the target domain.

2.4 Natural language processing

Being an interdisciplinary field that lies in the intersection of *artificial intelligence*, *machine learning*, *statistics* and *linguistics*, the aim of **natural language processing** (NLP for short) is to make natural language utterances processable by machines. Keyphrase extraction – which seeks to identify important phrases in unstructured textual documents – thus also belongs to the tasks of NLP.

Keyphrase extraction – and NLP tasks in general – can greatly profit from the machine learning techniques briefly described above. Here, we enumerate those important NLP tasks that are also related to keyphrase extraction. Comprehensive descriptions on these (and further) NLP tasks can be found in the literature on this area, including [52, 63, 70, 71].

2.4.1 Text classification

Text classification clearly illustrates the connection between machine learning and NLP as it is a classification problem where data instances to be classified come in the form of textual documents. The common example of text classification is the categorization of e-mail massages as either *spam* or *non-spam* (also called ham). This kind of task is a clear example of binary classification, whereas classification performed on the benchmark dataset of Reuters news articles [62] belongs the area of multi-class classification, since the task here is to label documents using several categories.

A common approach for classifying documents is to take all the *n-grams* (consecutive tokens) of some text as features and perform the classification based on them. It is obvious that keyphrases can also serve as features for representing documents because they can be regarded as exactly those phrases which characterize the main contents of documents in a condensed form. One possible advantage of doing so is that we can keep the most important aspects of a document (again by definition), yet characterize documents with several orders of magnitude less features, which can help us to overcome issues such as the much dreaded *curse of dimensionality*, being extensively discussed in the machine learning literature (see e.g. [15, 33, 47, 82, 83]).

2.4.2 Information Retrieval

A detailed description of **information retrieval** (IR for short) and the techniques applied in this area can be found in various resources, including [63, 71]. The most typical task of IE is that given a (possibly large) set of documents, we need to find those relevant documents that are capable of meeting certain information needs.

As document collections often contain unstructured and textual data, this task has a clear connection with NLP, since effective information retrieval systems must be prepared to handle linguistic phenomena such as inflexion – not only during the so-called indexing of the documents, but also during the processing of the user queries. **Question answering** can also be viewed as a related area, since in order to automatically find answers to some question, we first need to determine the set of relevant documents.

Typical tasks of information retrieval include the effective indexing of documents, finding the latent semantic factors underlying the data – a task outlined in Section 2.2.2 – e.g. with Latent Semantic Analysis (LSA) [61] and being able to effectively search in the (possibly semantically) indexed document set. Search engines like Google, Yahoo!, Bing, Baidu and Yandex are typical applications of this area.

As an alternative indexing strategy (in contrast to exhaustive indexing), we might decide to index documents just for their keyphrases. Such an approach and its possible utilization will be described in Chapter 6.

2.4.3 Summarization

The central application of **summarization** is to produce shortened versions of documents, i.e. their summaries. These summaries may be based on single documents or multiple documents at a time. One key approach employed in this task is to extract core sentences from documents, (see e.g. [95]). Such solutions can be augmented if the keyphrases of the documents are known, as the core sentences of a document are most likely those that contain especially relevant phrases for the document. Representing a document by its set of most important keyphrases can also be interpreted as providing a summary for it.

Chapter 3

Keyphrase Generation from Newswire

The aim of this chapter is to introduce an automated approach capable of assigning keyphrases to news articles. This problem can be interpreted as a specialized keyphrase generation task, where the determination of keyphrases is not exclusively restricted to such phrases that are present in the documents [72]. These keyphrases can be useful in organizing, retrieving and linking different news contents. In this chapter, we introduce our automatic keyphrase generation solution, especially designed for on-line news archives.

3.1 Motivation

In February 2009, the Hungarian news portal Origo introduced the manual assignment of keyphrases of their newly created contents. Here, we will list the major benefits and results, the utilization of keyphrases produced for the news site Origo, alongside with the special characteristics of the task which need to be taken care of when we wish to build a system which automatically performs the assignment of keyphrases to news articles.

The so-called channel pages provide a proxy for accessing articles that are related to each other by their contents. In fact, channel pages can be viewed as a realization of a multi-label classification of news articles into several categories. Such channel pages used to exist prior to the utilization of keyphrases, but these were created by the editors of the news portal. This resulted in the fact that only a very general-level topic hierarchy of broad and heterogeneous topics (including channels such as "*Sports*", "*Homeland*" or "*Business*") was maintained. With the utilization of keyphrases, a more detailed hierarchy of channels could be realized.

Generally speaking, assigning keyphrases to news contents involves the possibility of creating a number of channel pages that focus exclusively on a topic determined by some keyphrase. With the help of keyphrase-augmented channel pages, articles sharing a common keyphrase like "*financial crisis*" can be displayed next to each other. Obviously, channel pages can be created based on the

results of some Boolean operations, performed on multiple keyphrases, e.g. a channel page might only contain the kind of articles to which the keyphrases "*financial crisis* `AND` *Hungary*" are both assigned.

Note that similar queries can be formulated with standard information retrieval techniques, without relying on keyphrases (see the term *indexing* in [71]). However, performing queries based on (the composition of) keyphrases assures certain benefits due to the fact that ordinary index terms and keyphrases differ from each other in certain important aspects:

1. index terms are all the terms that are present in a document, irrespective of their relevance to a document, whereas keyphrases are only those terms that are relevant for the main topics of a document,

2. index terms are derived from the analyzed document itself, whereas keyphrases can also be such ones that are not present in a document.

The thematic range defined by a regular (i.e. not relying on keyphrases) channel structure is obviously very limited; with such channels only large and rather heterogeneous user segments (e.g. users interested in *sports* rather than users interested in *Eastern European football scandals*) can be created for behavioral targeting. The analysis of user habits can also be performed on a much finer grade, once keyphrases have been assigned to news articles.

One of the benefits of employing keyphrases is that it can also support targeted contextual advertisements. Provided that advertisements are also assigned key concepts to (possibly by the advertisers themselves), advertisements can be published in a highly relevant context by simply measuring the content of overlap between the key concepts of advertisements and news articles.

In parallel with the introduction of the assignment of keyphrases to news articles, a contextual advertising system was under development at Origo Ltd., which produced outstanding preliminary test results: for contextually published ads, the click-through-rates (being the ratio of how many times an advertisement is displayed and clicked on) increased by a factor of 10 for advertisements published on a contextual basis.

Keyphrases can also be useful for the linking of multi-modal contents. For example, videos can be recommended for news articles and articles can be also recommended for videos. These recommendations are straightforward if each type of content is assigned with keyphrases. Keyphrase extraction can also be applied for the transcripts of video speech and for the caption of images.

We should add that with the introduction of manually assigning keyphrases to the news articles, Origo could only partially enjoy the benefits previously described, as no tags were assigned to the news articles that were created before February 2009 (i.e. since the manual assignment of keyphrases to news contents is employed). As the manual assignment of keyphrases to the entire news archive would be a laborious task, means to automatize it was much desired.

Scientific publications of some research field – being the primary targets of standard keyphrase generation tasks – are more homogeneous than news articles from various aspects, including their topical diversity, writing style and structure. The typical heterogeneity of news articles pose certain difficulties when it comes to the generation of keyphrases from documents of this genre. These characteristics lead to the need for special approaches during the treatment of the task. Next, we will introduce our framework for the automatic generation of keyphrases to news articles.

3.2 Keyphrase Generation Framework

Our main approach sought to find the best set of keyphrases for the articles in the news archive. It consisted of three steps, namely

1. extracting a set of keyphrase candidates from the newswire documents, based on their linguistic analysis
2. extending the set of keyphrase candidates by exploiting semantic knowledge derived from Wikipedia
3. reducing the size of the extended set of keyphrase candidates via their ranking.

3.2.1 Generation of keyphrase candidates

As a first step, key concepts being present in the articles were gathered as a set of candidate keyphrases. Expressions that might behave as key concepts were defined as the names of people, locations and organizations – often playing central roles in news articles – and noun phrases.

Named Entity Recognition and lemmatization

We treated the standard classes of Named Entities (NE) (i.e. *organization* (ORG), *person* (PER), *location* (LOC) and *miscellaneous* (MISC)) as one possible source of keyphrase candidates as such phrases often play decisive roles in news articles. We trained Conditional Random Fields (CRF) [60] sequence classifiers that utilized the rich feature set for Hungarian NE Recognition [106] (NER for short) in order to determine NEs within the articles.

As the domain of news collections usually covers a wide spectrum, we trained different CRF classifiers to cope with the differences across domains in the task of NER. We were able build topic-specific NER models, as articles were assigned the meta-data relating to the thematic channels they belonged to. More information about this meta-data (i.e. the so-called channels) can be found below in Section 3.3.1.

During the extraction of NEs from the articles, the NER model corresponding to the channel information of the articles was employed. We applied a topic-ignorant NER model for those articles,

the topics of which was not among the topics that we built topic-sensitive NER models for. The topic-ignorant NER model was simply trained on all the annotated sentences being merged together, irrespective of their topics.

Since the NER models only identified the possibly affixed running text occurrences of named entities, we had to bring them to a normalized form before treating them as proper keyphrase candidates. When normalizing NEs, we followed a two-step strategy, including the lemmatization of NEs and the resolution of abbreviations.

In morphologically rich languages like Hungarian, nouns (including NEs) can have hundreds of different forms owing to grammatical number, possession marking and grammatical cases. When looking for the lemmas of NEs, the word form being investigated is deprived of all of the suffices it may bear. However, there are some NEs that end in an apparent suffices (such as '*McDonald's*' or '*Phillips*' in English). The difficulty of proper name lemmatization lies in the fact that – unlike common nouns – NEs cannot be listed exhaustively, due to their diversity and steadily increasing number.

The heuristic assumption that we employed during the lemmatization of NEs was that the lemmas of NEs have higher relative frequencies compared to their affixed forms. Hence, in order to be able to select the appropriate lemma for each NE phrase, we applied the following strategy: endings that seemed to be possible suffices were removed from the NEs; then the frequency of all possible lemmas in the news archive was counted and a decision was made based on these frequencies, employing rules learned from previous NE lemmatization experiments [37]. Next, we performed abbreviation resolution in order to avoid synonymous entities to be regarded as keyphrase candidates (e.g. by regarding both the NEs "*United Nations*" and "*UN*" as keyphrase candidates).

Extraction and derivation of noun phrases

Apart from Named Entities, common nouns and noun phrases (NPs) can often serve as useful keyphrases. We first attempted to extract all the NP chunks from sentences by relying on a constituent parser for Hungarian [1]. The constituent tree-based approach for the extraction of NPs proved to be considerably slow, and the noun phrases that we were able to define with the help of simple rules, based on the less resource-intensive morphologic analysis of the sentences, were of similar quality.

For the above reasons, we treated noun phrases as those tokens and sequences of tokens that consisted of some (possibly zero) adjectives, followed by at least one noun, i.e. matched the POS-pattern defined as `ADJ* NOUN+` (the * and the + symbols referring to the Kleene star and plus operators, respectively). A morphologic analysis of the sentences was conducted by the transformation-based learning (TBL) POS tagger [59] that was trained on the Szeged Treebank.

Apart from extracting noun phrases that were present in the news articles themselves, we also tried to generate NPs from verb phrases (e.g. "*the bank was robbed*" → "*bank robbery*"), adjectives (e.g. "*Italian*" → "*Italy*") and other NPs (e.g. "*the price of oil*" → "*oil price*"). A handful of

linguistically motivated transformation rules were applied during the generation of NPs.

To compensate for the possible shortcomings of the previous approaches concerning their recall scores, gazetteers containing entities and possible topic identifiers were compiled from Wikipedia. The phrases being present in the gazetteers were automatically treated as keyphrase candidates if they were found in some new article. The gazetteer contained all the titles of the Hungarian Wikipedia articles and the contents of all those Wikipedia articles, the titles of which started with the expression "*List of*" (e.g. "*List of African dishes*").

Ranking of the keyphrase candidates

Keyphrase candidates were extracted from the articles in the manner described above. Afterwards, we needed a way to rank these candidate terms and assign the highest ranked keyphrases to the individual articles, according to the expectations of Origo. These expectations were defined in their keyphrase assignment manual (see Section 3.3.1 for details). In order to perform a ranking of the candidates, we relied on their structural traits; for example if they were present in the headline of an article or were formatted as a bold or italic. Each structural trait was assigned an importance weight, which was then incorporated in the calculation of the relevance of the candidates. The relevance metric that we employed was a parameterized generalization of the $tfidf$ measure, having the form

$$tfidf(keyphrase) = \frac{(\sum_{type} \lambda_{type} * tf(keyphrase, type))^{\alpha}}{df(keyphrase)^{\beta}},$$

where $tf(keyphrase, type)$ refers to the keyphrase frequency of $keyphrase$ as a given $type$ (e.g. bold), $df(keyphrase)$ is the number of documents, for which $keyphrase$ is treated as a candidate, and the values of α, β and λ are hyperparameters to be optimized.

In order to find the optimal values for the parameters α, β and λ, we those articles which had keyphrases manually assigned to them by their authors (for more details on these documents see Section 3.3.1 which introduces the dataset we had access to). The values for these hyperparameters were determined such that those candidates should be ranked high which were also assigned to the articles by human indexers. We performed a grid search to determine the values of these parameters. The weights obtained by the grid search implied that the most important parts of the articles, where manually assigned keyphrases tend to be located, are the titles, headings, captions of images and the texts formatted in italics.

3.2.2 Keyphrase assignment based on Wikipedia

Phrases that can be derived from the documents themselves form only a part of the appropriate keyphrases, as there is a non-negligible set of keyphrases that do not occur explicitly in the documents. For example, an article dealing with the "*economic crisis*" may not contain the expression itself at all, but words, such as "*bankruptcy*" and "*recession*", being present in the document might

imply the appropriateness of that keyphrase. Such examples suggest that the pure extraction of keyphrases based on the contents of the documents might not always be sufficient, as the required keyphrases might not actually be present in the articles themselves. We will refer to keyphrases that are assigned to articles in such a manner as *abstract keyphrases* and the process of (automatically) determining abstract keyphrases to documents as (abstract) *keyphrase assignment*.

Our modules responsible for the assignment of the so-called abstract keyphrases, receive the set of keyphrase candidates extracted from the news articles themselves (as described in Section 3.2.1) and return a set of keyphrases, potentially behaving as proper abstract keyphrases for the news articles in question. Next, we will present those mechanisms that were responsible for generating the potential set of abstract keyphrases, given a set of non-abstract keyphrase candidates.

As a first step, the assignment of our keyphrase candidates to Wikipedia articles was carried out. We assigned a keyphrase candidate to some Wikipedia article, if the normalized title of the article was the same as the keyphrase candidate. When a keyphrase candidate was ambiguous (i.e. it had a disambiguation page on Wikipedia), we did not choose any of the Wikipedia articles possibly related to it, as we wanted to avoid improper assignment of keyphrase candidates to Wikipedia articles. Then, five different abstract keyphrase assignment methods, based on the recognized Wikipedia articles, were applied. These methods made use of both the textual contents and the rich link structure of Wikipedia. Next, we will present the five Wikipedia-based approaches one by one.

Consideration of redirect pages

The structure of Wikipedia enables the same contents to be accessed under different article names. For example, if we search for the term "*United States*" or "*Americans*", we get the same results. The pages responsible for redirection (redirect pages) can be utilized to find synonyms (e.g. "*United States of America*" - "*United States*"), create associations (e.g. "*American*" - "*United States*") and resolve acronyms (e.g. "*USA*" - "*United States*"). Based on these, we can determine a canonical representation of concepts, which has the benefit of increasing the cohesion of the keyphrases extracted at the corpus-level. In order to do so, we replaced those keyphrase candidates that were involved in a redirect relation. The (potentially abstract) phrase such a keyphrase candidate was replaced to was the title of the Wikipedia article its redirect relation pointed to.

Extraction of definitions

Owing to the encyclopedic nature of Wikipedia, articles usually start with a definition of the concept they describe. In order to extract definitions, we first determined the sentence which was the most likely to contain valuable definitions. In our approach, we treated the first sentence which also contains the title of the Wikipedia article as the one which might define the subject of the Wikipedia article. For a Wikipedia article which did not have its title written down in it, we selected its very first sentence as the one which might contain a definitional sentence.

Entity	Definitional sentence	Definition stump
Gottlob Frege	Friedrich Ludwig **Gottlob Frege**, German mathematician, logician, philosopher, the founder and researcher of modern mathematical logic and analytical philosophy.	Mathematics German Philosophy
Pál Erdős	**Pál Erdős** was one of the most outstanding mathematicians of the 20th century and the member of MTA.	Mathematics
The Sopranos	**The Sopranos** is an American TV-series, the creator and producer of which is David Chase.	American TV-series TV-series producer

Table 3.1: Example definitional sentences and definition stumps derived from Wikipedia

Based on the above-described procedure, we extracted definitions for those keyphrase candidates that had some Wikipedia articles assigned to them. When more keyphrase candidates of a news article shared a common nominal phrase in their definitions, that nominal phrase was treated as an abstract keyphrase for the news article containing the keyphrase candidates.

This kind of definition generation was capable of defining hyponimous `IS-A`-kind of relations between keyphrase candidates and concepts. For instance, we could infer predicates such as *The Sopranos* is an *American TV series*. Suppose that an article from the news archive was assigned multiple keyphrase candidates that were also defined as being American TV-series. In such a case, we could regard this as an indication that the particular news article was strongly related to American TV series.

Exploiting the above observations, we extracted all the possible nominal definition stumps from the definitional sentences that we derived from Wikipedia articles. These definition stumps acted then as abstract keyphrase candidates. During the extraction of these definitional nominal phrases, we relied on morphosyntactic rules (e.g. we selected the noun occurring next to the first mention of the title of a Wikipedia article as a possible definition phrase). Next, definition stumps were regarded correct either if all the tokens constituting the nominal phrase or the entire phrase itself could be mapped to some Wikipedia article. This criterion served the purpose of increasing the precision of the definition stumps generated. Table 3.1 contains examples for both definitional sentences and nominal structures that were derived as abstract keyphrase candidates for various entities.

Utilizing the link structure

We also examined the possibility of assigning abstract keyphrases by exploiting the rich link structure of Wikipedia. Here, we employed the following three metrics:

1. we looked for those Wikipedia articles which frequently co-occurred on Wikipedia with some keyphrase candidate in the form of anchor texts (referred to as the *Co-occurrence* method in our evaluations),

2. we examined those Wikipedia articles that were frequently referred by such Wikipedia articles

that were assigned to the set of the keyphrase candidates of news documents (referred to as the *Outgoing links* method in our evaluations),

3. we also looked for the titles of Wikipedia articles, the contents of which showed substantial overlaps with the sets of keyphrase candidates derived from news articles (referred to as the *Article relatedness* method in our evaluations).

In the case of examining co-occurrences, we looked for the titles of those Wikipedia articles that were frequently co-mentioned as an anchor text on Wikipedia with some keyphrase candidate. This metric was utilized in such cases where the Wikipedia article corresponding to a keyphrase candidate had at least 10 but no more than 150 occurrences on Wikipedia in the form of a hyperlink. We did this because those articles that were referred fewer than 10 times seemed to be of low relevance, while those referred more than 150 times were too general.

For articles falling in the above-mentioned range (with respect their mentions as links), we looked for those Wikipedia articles that were accompanying their mentions as a link in at least 50% of the cases. As for an illustrative example, since the co-occurrence measure for rally racer "*Sébastian Loeb*" and "*rally world championship*" turned out to be 0.7073, the latter term was also applied as an abstract keyphrase for those news articles for which "*Sébastian Loeb*" was extracted as a keyphrase candidate. Note that from an associational rule mining point of view, we could rephrase the aim of the co-occurrence method as finding such association rules which had keyphrase aspirants on their left-hand side with an absolute support between 10 and 150 and a confidence score exceeding 0.5.

When examining outgoing links, we looked for Wikipedia articles that could be considered as relevant to a set of keyphrase candidates. We took every Wikipedia article that were referred by reliable outgoing links from those Wikipedia articles that were corresponding to some set of keyphrase candidates of a news article. We treated an outgoing link of an article as reliable if the article referred by it contained a back-reference to the referrer article, or if at least 25% of the links of a referring article pointed to the same article, and the number of the links was more than 3.

At the document level, a Wikipedia article considered reliable and its title was used as an abstract keyphrase if it was regarded as a reliable link for more than one Wikipedia article associated with the keyphrase candidates of some news article. For example in the case of an article which contained both "*BUX*" and "*Stock Exchange of Budapest*" as candidates, the use of the abstract keyphrase "*economy of Hungary*" was inferred, due to the fact that Wikipedia articles associated with the former two terms contain reliable outgoing links to the latter Wikipedia article.

As a third way of generating abstract keyphrases, we inspected the outgoing links of Wikipedia articles. This submodule chose the titles of those Wikipedia articles for which the outgoing links showed a substantial overlap with W, the a set of keyphase candidates with Wikipedia articles being assigned to them. The relatedness score calculated for a particular Wikipedia article d_j was obtained

	automatic	manual
Start date	5/12/1998	15/2/2009
End date	14/2/2009	22/10/2009
Number of articles	366,937	28,055
Number of keyphrases created	66,843	15,726
Number of new articles created (daily average)	93	110
Number of new keyphrases (daily average)	17	45
Average number of keyphrases per article	4.98	3.42
Average length of keyphrases used (tokens)	1.45	1.48

Table 3.2: Statistics of the manual and automatic keyphrase assignment

using the formula

$$ArticleRelatedness(d_j, W) = \frac{|W \cap o(d_j)|}{|o(d_j)|} \frac{\sum_{t_i \in W} tfidf(t_i, d_j)}{|W|},$$

where $o(d_j)$ is the number of outgoing links from article d_j. The first term of the formula penalizes Wikipedia articles acting as hubs, i.e. containing a large collection of links to other pages. We introduced this penalty term as those Wikipedia articles acting as hubs have a bigger chance of containing a substantial amount of the elements of some W (simply due to the excessive number of links such pages contain), nevertheless, such hub pages are also less valuable, due to the presumable generality of the concept they describe. The second part of the formula is then the average of the $tfidf$ scores of the elements in W with respect d_j. The bigger this term is, the more relevant terms from W are contained in the Wikipedia article d_j in the form of links. In the end, we treated the title of a Wikipedia article d_j as an abstract keyphrase, if the inequality $ArticleRelatedness(d_j, W) > 0.3$ determined via a development set held.

3.2.3 The final set of keyphrases

After extracting keyphrase candidates from the articles and extending them with abstract keyphrases, the average number of keyphrase candidates per document was 17.3. This value was substantially higher than required, i.e. 5 keyphrases per article. Hence there was a need for filtering the sets of keyphrases assigned to the articles. The selection was performed based on the ranking of the keyphrase candidates as described in Section 3.2.1. Note, however, that the parametrized tf-idf score could not be calculated for abstract keyphrase candidates, as the formula involves the within-article frequency of the keyphrases (which is 0 for abstract keyphrases by definition). For this reason, abstract keyphrase candidates were sorted based on their overall frequency in the news archive and the number of times they were proposed as being abstract keyphrases. As can be seen from Table 3.2, the post-processing had the effect of reducing the average size of the set of keyphrases per document to the desired value.

Channel name	Number of documents	Size (%)
Sports	72,018	18.91
Homeland	57,012	14.97
Abroad	49,689	13.05
Business	49,662	13.04
Technology	19,274	5.06
Entertainment	15,626	4.10
Cars	12,933	3.40
Other	117,578	27.47
Total	380,859	100.00

Table 3.3: Frequency statistics of the largest topmost-level channels

3.3 Experiments and discussion

Here, we give a detailed description of both the news archive dataset and the keyphrase assignment manual that the employees of Origo provided for us. Following that, we report our experiments aiming at the evaluation of our automatic keyphrase assignment system.

3.3.1 Dataset

Origo is one of the most visited news portal in Hungary, reaching about 45% of all the Internet users in the country. The site was launched in December 1998 and over 380,000 articles were published by the end of 2009, making Origo admittedly the owner of one of the largest and most diverse digital news archives in Hungary. As a typical general interest portal, Origo covers a very wide spectrum of topics and themes. Journalists at Origo started assigning keyphrases manually to their published contents from February 2009.

As we mentioned earlier, articles were organized into so-called channels. These channels were created by the employees of Origo and their role was to mark the main topic of the articles being assigned to them, such as *Sports* or *Technology*. The channels were divided into smaller subchannels, forming a hierarchic taxonomy in this way. During our experiments, we relied only on the topmost level of this hierarchic taxonomy, the most frequent channels of which are listed in Table 3.3. We should add that the channels were prone to changes over time: new (sub)channels could evolve over time, and some could even disappear. The reason why we relied just on the topmost level of the hierarchy was that this part of the taxonomy was more or less constant over the years.

As previously indicated in Section 3.2.1, we trained topic-specific NER models for the most common topics (i.e. channels) in the news archive. All together there were more than 20 top-level channels defined in the news archive, some of which had just a few articles assigned to them. For this reason, we decided to collect and annotate articles belonging to the most important channels in order to make the training of channel specific NER models possible. We decided to train channel-specific NER models for the six largest channels besides the channel *Business* (as we already had

Channel name	#NEs	#NEs per 100 tokens	avg. token length of NEs
Sports	15,500	8.76	1.46
Homeland	1,995	4.63	1.62
Abroad	3,295	5.07	1.41
Technology	4,488	4.08	1.68
Entertainment	7,669	6.47	1.80
Cars	9,519	4.77	1.45
Merged	42,466	5.95	1.54

Table 3.4: Statistics of the NER training corpora

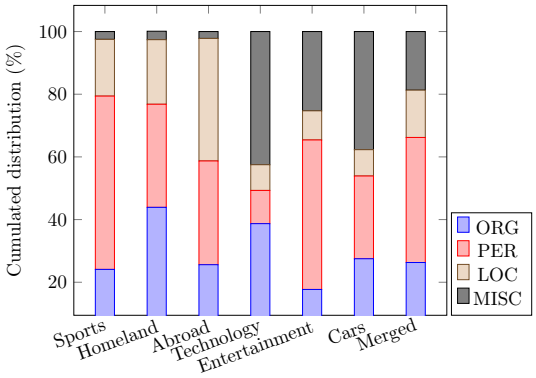


Figure 3.1: Relative frequencies of NE-types in the NER training corpora

a reliable NER model available on the business domain). A detailed distribution of the sizes of the top-level channels is given in Table 3.3.

During the annotation of the articles that we sampled from the six largest channels, we found that named entities originating from different topics behave fundamentally differently, as we had supposed. Various statistics on the distribution of NEs in different channels can be seen in Table 3.4 and Figue 3.1. From Figure 3.1, we can see how different the distribution of NEs among the different channels actually is. Table 3.4 and Figue 3.1 also contain information on the merged training data, which were derived from the union of the channel-specific training data.

Keyphrase assignment manual

The principles on how Origo employees should perform keyphrase assignment are summarized in a guideline. During the design of our system, we had access to this document and our goal was to produce the kind of automated keyphrase assignment system which strictly confirmed with the principles stated in this document.

The guideline defined four types of keyphrases, namely *topic*, *person*, *organization* and *location*. We needed to assign at least one keyphrase of type *topic* for each article. The guideline also stated that only entities that play a decisively important role in some article should be chosen as keyphrases.

The guideline included the best practice with respect to what should be applied as a keyphrase (by providing typical examples), and what should be avoided (e.g. slang, metaphors, paraphrases, verbs, adjectives and pronouns). The document also defined the ideal number of keyphrases for different journalistic genres and made clear what kind of expressions were too general or too specific to act as a keyphrase. The guideline also offered advice on

1. how to avoid collisions with keyphrases which would have the same surface forms, but multiple different meanings (e.g. to use keyphrases such as "*László Kovács the politician*" and "*László Kovács the boxer*" instead of using the single keyphrase "*László Kovács*") and

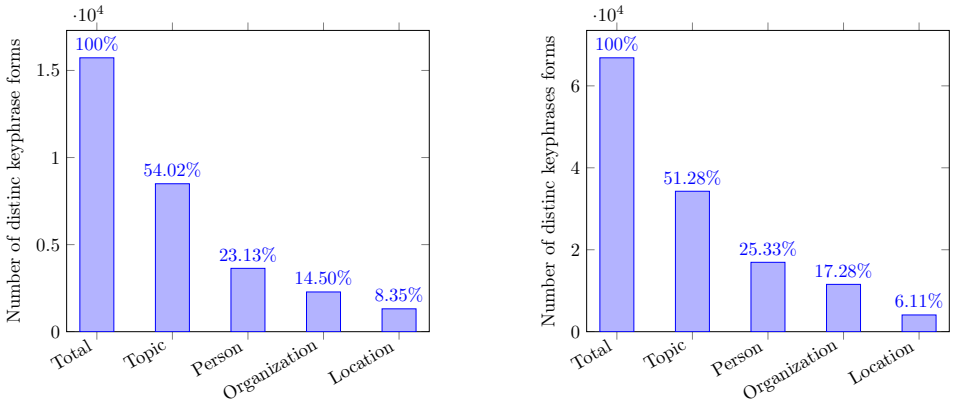
2. how to avoid the creation of multiple keyphrases for the same meaning (e.g.to use consistently one of the keyphrases *H1N1*, *Swine Flu* or *Influenza A virus*, instead of applying them simultaneously).

The editorial system suggests keyphrases to the employees by auto-completion, but users are free to use previously unused keyphrases. Suggested keyphrases are based both on existing keyphrases and on the Comprehensive Hungarian Thesaurus [114] containing 21 thousand items. Suggested keyphrases have the main benefit of keeping the whole set of keyphrases on the portal consistent, as both conceptual divergences (exploiting synonyms from the thesaurus) and spelling variants can be avoided. Suggesting keyphrases, however, can have a detrimental side-effect as users might want to quickly pass their duties and accept keyphrases suggested by the system, without really deciding on the appropriateness of the auto-suggested keyphrases.

3.3.2 Evaluation

We decided to apply the state-of-the-art keyphrase extraction system, KEA[119] as an alternative automated approach to perform the assignment of keyphrases to the news articles in the archive. As the framework was originally intended to handle documents written in English, we made it able to handle features being specific to Hungarian. KEA was then finally trained on the subset of the archive which had keyphrases assigned to them by their authors (i.e. that were created after 15/2/2009), serving as gold standard keyphrase assignments. The framework obtained this way served as our baseline solution for the keyphrase assignment task.

In order to provide quantitative performance scores for the baseline system and our proposed solution, 725 articles were randomly selected from the news archive. The automatically defined sets of keyphrases by both systems were then inspected by employees of Origo Ltd. The scoring schema employed during the manual evaluation of the test articles was established by the employees of Origo prior to the human evaluation they performed. Due tp the scoring schema each automatically generated keyphrase needed to be assigned either a (positive) reward or a (negative) punishment score. Table 3.5 contains the results achieved by our framework and that of KEA, the latter being our baseline.



(a) Keyphrases assigned manually (b) Keyphrases assigned automatically

Figure 3.2: Distribution of the types of manually and automatically assigned keyphrases

Method	Precision	Recall	F-score
Baseline	22.64	54.86	**32.05**
Proposed	59.42	91.26	**71.92**

Table 3.5: Results achieved by the automatic keyphrase generation systems

Apart from the performance at individual keyphrases level, we were also interested in the quality of the automatically determined keyphrases at the document level. In order to derive article-level decisions on the sets of keyphrases determined for the news article in the test set, the keyphrase-level scores were added up for each document. When the overall score for an article was positive, it was interpreted as the overall quality of the set of keyphrases being satisfactory. Using this evaluation metric on the randomly selected 725 articles, our system produced acceptable sets of keyphrases for 75.44% of the test documents, incontrast to 19.03% for KEA.

We also analyzed some of the characteristics of the keyphrases assigned manually by the content creators and our automated approach. These statistics are summarized in Table 3.2 and Figure 3.2. Figure 3.2 shows for instance that the distributions of the keyphrase types for the automatically generated keyphrases and the manually assigned keyphrases matched each other closely.

As regards the evaluation of the abstract keyphrase assignment submodule, we also performed a human evaluation. For these experiments, two linguists were hired to decide on the appropriateness of each keyphrases assigned to the news articles. 600-600 news articles were chosen for evaluation, out of which 100 were the same for both annotators; resulting in a total of 1,100 documents for the evaluation. The original authors of these articles assigned 1,114 abstract keyphrases to them. The abstract keyphrase assignment procedure enhanced by Wikipedia produced all together 5,014 assignment of 2,028 distinct abstract keyphrases for the test documents.

The procedure of evaluation was as follows: annotators had to examine each abstract keyphrase

Method	Abstract keyphrases induced	Precision
Redirections	1,155	72.38
Definitions	1,471	28.14
Co-occurrence	1,998	34.88
Outgoing links	558	40.68
Article relatedness	551	16.33
Overall	5,733	39.49

Table 3.6: Results achieved by different abstract keyphrase assignment heuristics

	Precision	Recall	F-score
Annotator #1	39.33	10.57	16.66
Annotator #2	38.48	10.77	16.83
Aggregated	38.91	10.67	16.75

Table 3.7: Results achieved by abstract keyphrase assignment enhanced by Wikipedia

assigned to an article, and decide whether it was an acceptable keyphrase with respect to the content of the document (precision), taking the keyphrase assignment manual of Origo into account as well. At the same time, they had to decide whether the abstract keyphrases automatically assigned to the documents were able to semantically cover the meaning of one or more abstract keyphrases that were assigned manually by the editors of Origo (recall). The precision scores along with the number of abstract keyphrases generated by each method are present in Table 3.6.

The overall performance scores for the two independent annotators and the aggregation of their decisions on the quality of the abstract keyphrases are reported in Table 3.7. F-scores combine the precision of automatic abstract keyphrases and the extent to which abstract keyphrases were able to cover the abstract keyphases manually assigned to the test documents. These results are satisfactory, if we take into account the fact that coverage was compared to the keyphrases manually assigned by the employees of Origo, who had (as human beings) access to a full sense repository and not just Wikipedia (where only 20.76% of the manually assigned abstract keyphrases had a corresponding Wikipedia article).

3.4 Related work

The fact that there has been a handful of research aiming at the processing of news data clearly indicates the importance of the task described in this chapter. Here, we list the most related studies to ours.

Grineva et al. [43] proposed an unsupervised approach, utilizing Wikipedia and the modularity maximizing graph partitioning algorithm of Newman and Girvan [87]. In their approach, a semantic graph was constructed, the nodes of which were terms and the weighted edges between them were reflected the strength of the semantic relatedness between them. Keyphrases were then derived from

the communities, i.e. the partitions of that graph. During their evaluation, they experimented with noisy and multi-theme documents (such as blog and news contents), and they reported an F-score of 40.3 on news documents in English.

Neto et al. [86] introduced their media monitoring solution which was a pipeline of several processing modules, capable of keeping track of broadcast news streams. In their study, they handled several languages, i.e. English, Spanish and Portuguese (both European and Brazilian). This system differed from ours as it handled audio data and its transcriptions, while the data that we processed was originally written by journalists.

Marujo et al. [73] demonstrated the applicability of the processing of broadcast news by building a keyphrase cloud generation system from the extracted keyphrases. In their later study, Marujo et al. [74] also introduced their topical keyphrase extraction framework, which relied on crowdsourcing for the collection of topical keyphrases. In their other study, Marujo et al. [75] realized the importance of Named Entities as we did, and they introduced their solution for the extraction of Named-Events (as they called it) from news documents, again by relying on crowdsourcing.

In their study, Marujo et al. [76] proposed the filtering of news documents. They showed how the performance of the supervised keyphrase extraction framework called Maui [79] could be improved when certain sentences of the news articles were filtered prior to keyphrase extraction.

Others, like Ding et al. [32] employed binary integer programming in the extraction of keyphrases from news articles and Wan and Xiao [117] utilized the neighborhood information residing in the similar documents for extraction of keyphrases from newswire. Besides using their framework on scientific literature, Bougouin et al. [19] also experimented with their unsupervised graph-based keyphrase extraction system on news articles derived from the French version of WikiNews. On that dataset, they reported an F-score of 35.6.

Not related to the processing of news documents, experiments have been conducted for extracting definitions and various semantic relations between entities based on Wikipedia. These lines of research relate to ours, as we used similar approaches to provide solutions for the assignment of abstract keyphrases to news articles. Ye et al. [122] used Wikipedia to derive definitions by relying on positional information and the contents of the infoboxes of the Wikipedia articles. Other studies also attempted to derive valuable information, taxonomies and ontologies with the help of Wikipedia, see e.g. [93, 103, 120].

3.5 Summary of the thesis results

In this chapter, the author introduced a novel framework for the automatic assignment of keyphrases to news articles. Related to his publication [38], the author regards the following as his main contributions to the research topic:

1. Ranking procedure for selecting the most likely keyphrases of news articles A parametrized generalization of the $tf - idf$ score was introduced which was able to rank higher those keyphrase candidates that human indexers would select for the news articles. The metric takes into consideration positional traits and characteristics related to the formatting of keyphrase candidates when deciding on their relative importance among the set of candidate phrases. These features turned out to be useful when deciding on the importance of keyphrases of newswire documents.

2. Assignment of abstract keyphrases based on definitions derived from Wikipedia The proposed procedure to retrieve definitions from Wikipedia is capable of defining hypernymous relations between entities. Relying on the knowledge extracted from Wikipedia that way, an efficient method for assigning abstract tags to documents was suggested as well.

3. Assignment of abstract keyphrases based on the link structure of Wikipedia Methods for exploring further semantic relations between entities were also introduced. These methods were incorporating the knowledge encoded in the link structure of Wikipedia. Not only association rules were generated from the links of concepts which frequently co-occurred on Wikipedia, but the outgoing links of Wikipedia articles were also utilized during the generation of abstract keyphrases. Relying on these ideas, it became possible to assign useful abstract keyphrases to documents.

Chapter 4

Keyphrase Extraction from Scientific Documents

In this chapter, we present a feature rich keyphrase extraction framework for the processing of scientific documents. Due to our quantitative analysis, the proposed model performs competitively or even better than other state-of-the-art approaches.

The nature of the approach applied in this chapter differs fundamentally from the one presented in Chapter 3 for multiple reasons. Firstly, we introduce a model here that deals with documents written in a different language (English instead of Hungarian) and which belong to a different domain (scientific literature instead of newswire text). Also in this chapter, we aim at the task of keyphrase extraction (not performing keyphrase assignment), meaning that we do not intend to determine such keyphrases that are not present in a document (which we referred to as abstract keyphrases in Chapter 3). Another difference is that in Chapter 3, we had access to a large and heterogeneous set of documents, whereas we will rely on relatively small (training and test) sets of documents from a well-defined topical scope throughout this chapter. Because of this, here we focus on the design of supervised models for the extraction of keyphrases.

4.1 Motivation

Becoming familiar with a research field or just simply keeping up with the current results can be very challenging for experienced researchers as well due to the ever increasing amount of scientific literature available. As an illustration, the approximation in [16] says that in the year 2006 alone the number of scientific publications exceeded 1.6 million. Another approximation described in [51] says that the cumulated number of publications exceeded 50 million in 2009. Most of the similar studies agree that the average rate of increase in the number of journals ranges between 3% and

3.26%, which implies that the number of journals is expected to double every 22-24 years.

The extensive volume of scientific publications clearly motivates the usage of keyphrases for their easier retrieval. Even though there are journals and conferences which require authors to assign keyphrases to their publications, not all papers are accompanied with such valuable sets of phrases. Furthermore, different people might find different keyphrases appropriate for the same document. Thus automatic induction of keyphrases – based on personalized keyphrase extraction models – can still be a reasonable choice for documents with an existing set of author assigned keyphrases.

4.2 Keyphrase Extraction Framework

Here, we shall introduce a supervised machine learning approach for the extraction of keyphrases from scientific publications. In our framework, keyphrase candidates were extracted from the articles and those being present among the set of gold standard keyphrases were regarded as positive training examples. Using the notation defined in Section 1.1, the set of candidate phrases for document i, C_i is partitioned into two disjoint sets, $C_i^+ = C_i \cap K_i^*$ and $C_i^- = C_i \setminus C_i^+$, the former representing positive training examples, and the latter consisting of negative (i.e. improper) keyphrase candidates.

Maximum Entropy modeling was employed and the top-n keyphrase candidates with the highest posterior probability values of belonging to the class of gold standard keyphrases by the classifier were treated as keyphrases for a test document. Next, we will be describe how keyphrase candidates and the feature space representing them were constructed.

4.2.1 Generation of keyphrase candidates

One important aspect in keyphrase extraction is the way keyphrase candidates are selected and represented. As a high imbalance usually exists among the number of potentially extracted n-grams and the actual number of genuine keyphrases in a text, keyphrase candidates should be filtered instead of using any successive n-grams.

In our definition, keyphrase candidates were the n-grams that were not longer than 5 tokens and started and ended with a non-stopword token having one of the POS-tags of noun, adjective or verb. Phrases that fulfilled the above-mentioned criteria might still be discarded, due to their positional characteristics, i.e. a phrase was not treated as a keyphrase candidate if all of its occurrences were located in the *References* part of an article. It allowed us to prevent such phrases from becoming candidates which only had presences as part of some references.

Once we had generated the keyphrase candidates, they had to be converted to their normalized form. The normalization of an n-gram consisted of lowercasing and Porter-stemming each of the lemmatised forms of its tokens, then putting these stems into alphabetical order (while omitting the stems of stopword tokens). With this kind of representation, it was then possible to handle two

orthographically different, but semantically equivalent phrases, such as *diffusion of innovation* and *Innovation diffusion* in the same way, i.e. *innov diffus*.

4.2.2 Filtering of the candidate set

As mentioned previously, the number of potentially extracted phrases might exceed by orders of magnitude the number of genuine keyphrases that could be extracted from a given document. Treating keyphrase extraction as a supervised learning task and not paying enough attention to the selection of keyphrase candidates might result in gold standard keyphrases being underrepresented both during training and testing phase, which might affect adversely the automatic identification of keyphrases.

A possible way of overcoming this problem is to restrict the extraction of keyphrase candidates, i.e. filter them in such a way that as many genuine keyphrases as possible are turned into classification instances, while ruling out as many improper sequences of words as possible. The methods listed here contain stopword-based restrictions and the utilization of WordNet to incorporate semantic knowledge. The effects caused by our candidate phrase restricting policies are presented below.

Introducing stopword rules

Our initial definition of keyphrase candidates, (i.e. those n-grams which both start and end with either a *noun*, *adjective* or *verb*) did not say anything about the tokens with indices $n-1 \geq i \geq 2$ for n-grams of $5 \geq n \geq 3$. This restriction includes the elimination of those keyphrase candidates that were highly unlikely to serve as a gold standard keyphrase based on how often they contained stopwords.

The assumption here was that members of the gold standard keyphrases for a particular document tend to occur in it such that they have at least one occurrence without including any stopwords. As an illustrative example, the previously mentioned normalized form, *innov diffus* was discarded from the set of candidate phrases of a document if all of its occurrences had the form *diffusion of innovation* (i.e. containing the stopword *of* in all of its occurrences). However, if the normalized phrase had at least one single occurrence as *innovation diffusion*, then the normalized form was not discarded and became a keyphrase candidate.

With the help of this filtering process, normalized n-grams that were unlikely to act as keyphrases could be excluded from the set of candidate phrases, as it happened with the classification instance *basi method*, being the normalized form of the n-gram *basis of the method*.

Incorporating WordNet knowledge

Experiments were also carried out using WordNet [39] in order to provide an alternative way to normalize phrases. In this setting, the normalized form of a single token was determined by first searching for all its synsets (in the case of verbs, these were those noun synsets that were derived from

Filtering				
Stopword	WordNet	Instances	Positive instances	Negative instances
false	false	404,967	2,017	402,950
false	true	398,272	2,166	396,106
true	false	223,614	1,949	221,665
true	true	217,956	2,095	215,861

Table 4.1: The effects of filtering steps on the number of positive and negative training samples on the SemEval dataset [55]

the synsets of the verbal word forms). Next, instead of Porter-stemming the lemma of an original token, its most frequent word form was stemmed. The most frequent word forms were determined based on the estimated frequencies of WordNet for all the word forms among the synsets belonging to the original token (or their noun derivative synsets in the case of verbs). In this way, two word forms that originally would have been stemmed differently, such as *optimize* and *optimum*, could be stemmed to the same root forms. Another advantage of this procedure is that it is able to handle semantic similarity to some extent due to the fact that a word form is treated as if it were the most frequent word form among its synsets (e.g. the word form *task* is treated as if it were the word form *job*).

The effects of the previously mentioned filtering steps on the number of positive and negative training samples are summarized in Table 4.1. As we can be seen, the number of training instances nearly halved, without effectively modifying the number of positive training instances (in fact, when utilizing WordNet their number can also increase, due to its phrase normalization capabilities). This suggests that the proposed filtering technique is able to almost exhaustively rule out only those keyphrase candidates that were originally negative examples for being keyphrases.

4.2.3 Feature representation

To provide a baseline for our solution, we implemented the basic feature set of KEA [119] as it is one of the most cited publicly available tools for supervised keyphrase extraction. We did not use the KEA framework itself, as we employed a different strategy for generating keyphrase candidates, but rather reimplemented its basic features in our system. These features are the tf-idf score and relative first occurrence (i.e. the quotient of the first token position of a certain keyphrase candidate and the number of tokens in the document which contains it).

Our baseline solution also incorporated the use of standard deviation (expressed in the start token positions) of keyphrase candidates, which is also an optional feature in the KEA framework. This feature takes on smaller values if a keyphrase candidate is mentioned only at some specific section of a document and takes higher values when a keyphrase candidate is mentioned repeatedly at various points of a document. Phrases that are more important and might serve as a keyphrase tend to be used repeatedly, e.g. in the introduction and the conclusion parts as well.

Normalized candidate	Wikipedia article	Example Wikipedia category
result	Results	1989 albums Pet Shop Boys albums Epic Records albums
distribut hash table	Distributed hash table	Distributed data-storage File sharing

Table 4.2: Example categories the Wikipedia articles assigned to normalized candidate phrases belong to

Wikipedia-derived features

Wikipedia arguably provides a deep insight into human knowledge, which suggests that it could be used in the determination of keyphrases of scientific documents.

Utilizing Wikipedia categories One set of features was designed to make use of the taxonomy-forming category hierarchy of Wikipedia, indicating which articles belong to which (sub)categories. Candidate phrases were described by binary features indicating which categories the Wikipedia articles having the same normalized article title belong to. In the case a candidate phrase could not be aligned to any of the Wikipedia articles, none of the Wikipedia category structure-based features activated for that particular instance. Otherwise, those features fired that corresponded to the normalized nominal parts of the Wikipedia categories their aligned Wikipedia articles belonged to. The normalization of the Wikipedia category names was the same as that of keyphrase candidates (see Section 4.2.1). This way, approximately 10,000 new features were introduced. Note that when the normalization of the category names was not performed, three times as many features were introduced which could cause data sparsity issues for the model. Table 4.2 contains examples of features that were induced based on the category hierarchy of Wikipedia for two candidate phrases; the first one being more likely to act as a keyphrase than the other one.

Utilizing multiword expressions (MWEs) from Wikipedia Multiword expressions are lexical items that can be decomposed into single words and display idiosyncratic features [97], in other words, they are lexical items that contain spaces.

The fact that multiword expressions often turn out to be keyphrases implies that knowing which phrases are MWEs in a given text can be exploited in the determination of keyphrases. However, we should add that the two tasks (i.e. finding the MWEs and the keyphrases of documents) should be treated differently, since not all multiword expressions necessarily behave as keyphrases in every context (e.g. although the phrase *research group* is definitely an MWE, when it is present only in the affiliations part of a scientific paper, it should not normally be selected as a keyphrase).

To be able to decide which phrases might function as MWEs, a wide list of possible MWEs were collected from Wikipedia (using its 07-01-2011 dump): all the formatted (i.e. bold or italic)

and anchor texts of links from Wikipedia that was at least two tokens in length, starting with lowercase letters and contained only English characters or some punctuation, were collected. Having constructed that extensive list consisting of approximately 680,000 entries, an alignment of its elements and the corpus was carried out (taking into account linguistic alternations as well), treating those n-grams as genuine MWEs that started and ended with tokens of either a noun or adjective POS-tag and had no other (possibly zero) tokens in between them that were tagged as either a noun, adjective, preposition or possessive ending.

When deciding on the MWE-related features of a keyphrase candidate, we decided whether it

- was annotated by the automatic process (based on the MWE list extracted from Wikipedia and the POS-sequence of a candidate phrase, e.g. *maximal social welfare ratio*) as complete MWE,
- could be assembled from two MWEs of the list (e.g. *resource allocation problems*, where *resource allocation* and *allocation problems* were on the list separately, but not as one phrase),
- could be a superstring of at least one MWE (e.g. *general analysis remains*, due to the presence of *general analysis* on the list of MWEs).

Linguistic and orthographic features

As some POS-patterns are more characteristic of keyphrases, the authors of [50, 88] also proposed to derive features from the POS-tags of keyphrase candidates. Features generated by the POS-tags belonging to the tokens of different orthographic occurrences of a normalized phrase were applied in our study as well. Entire POS-tag sequences seem to be more informative compared to the simple indication of the presence of POS-tags in an n-gram, but it is also true that taking all the combinations of POS-sequences as a separate feature might invoke data sparsity issues.

To overcome this problem, we decided to add positional information to the POS-features derived from the individual tokens an n-gram consisted of. Features of POS-tags that were assigned to a token being itself a 1-token long keyphrase candidate, at the beginning, at the end and inside an n-gram, got a prefix of *S-*, *B-*, *E-* and *I-*, respectively. For instance, the phrase *dynamic/JJ semantics/NN* induces the features *B-JJ*, *E-NN* to fire, whereas the 1-token-long phrase *semantics/NN* induces the feature *S-NN* to do so. This way, POS-features were expected to contain probably less information, but to behave better with respect to dimensionality. In order to see the differences between the two approaches, both sequential and non-sequential POS-tagging feature representations were implemented and evaluated within the framework.

A set of binary features was implemented that was related to the orthography and semantics of keyphrase candidates, as Named Entities usually both have special orthographic characteristics and special semantic roles in their content. The position of NEs within candidate phrases was reflected in these features in a similar way as it was achieved for POS-tags: separate features were created to

indicate whether an n-gram contained a certain type of NE-class located at the beginning (B), inside (I) or at the end (E) of a keyphrase candidate. A special symbol for single token (S) keyphrases candidates was also reserved. For instance, the phrase *Nash* had the feature *S-PER* set to true, while *Nash equilibrium* had the feature *B-PER* set as true (and *S-PER* as false, naturally).

Keyphrases often have other special orthographic characteristics, e.g. it is the case with *UDDI* (being an acronym of the technical term *Universal Description Discovery and Integration*). Owing to the fact that not just the normalized, but the original forms of the candidate phrases were also stored in our representation, it was possible to construct two features for this. The first feature was responsible for character runs (i.e. more than 2 of the same consecutive characters), and another is responsible for strange capitalization (i.e. the presence of uppercase characters besides the initial one). The *I-*, *O-*, *B-*, *S-* prefixes were applied here as well, just like that for the Named Entity and POS features. Together with the NE-related features, these features formed those which are referred here to as *Orthography* features.

4.3 Experiments and discussion

In order to conduct experiments, we used two benchmark datasets to evaluate the quality of the keyphrases extracted from scientific documents. In the following, we will elaborate on these datasets, then we will present our experimental results achieved using them.

4.3.1 Datasets

During our experiments we used the dataset of the SemEval shared task on keyphrase extraction [55] and the Inspec dataset [50]. Here, we will introduce these datasets briefly.

SemEval shared task dataset

The primary dataset we used to test the effectiveness of our approach was the dataset of the SemEval-2 shared task on keyphrase extraction [55]. This dataset is a subset of the ACM Digital Library and consists of 244 scientific papers of length ranging from 6 to 8 pages taken from four different research areas (i.e. Distributed Systems, Information Search and Retrieval, Distributed Artificial Intelligence – Multiagent Systems, Social and Behavioral Sciences – Economics).

The set of documents was split into a training set of 144 documents and a test set of 100 documents by the organizers of the shared task. Sets of gold standard keyphrases assigned by both the readers and the authors of the publications in the dataset were included, which permitted the use of supervised learning. As the primary ranking criterion in the shared task was based on the evaluation against the reader-assigned keyphrases, we treated those phrases as the gold standard set of keyphrases during the training phase of our models. Other evaluations when the keyphrases were identified as the union of the author and reader-assigned phrases of the documents were also carried

out. We shall refer to the latter type of gold standard annotation as the combined one. We should also add that there was often a substantial overlap between author and reader-assigned keyphrases.

Inspec dataset

The other keyphrase extraction dataset we used for the evaluation of our approach is a subset of the Inspec database. It was originally created for the experiments described in [50] and it consists of 2,000 scientific abstracts with both controlled and uncontrolled sets of keyphrases identified by professional indexers. The elements of the controlled set of keyphrases are required to be present in a thesaurus of index terms, whereas uncontrolled keyphrases were terms freely assigned to articles by the indexers. The document collection was split into a training set of 1,000 abstracts and the development and test set each consisted of 500 abstracts. As we wanted to see the general applicability of our proposed model – which was primarily intended to perform well on the SemEval dataset – we simply discarded the development set and trained a model with the same settings as we did for the SemEval dataset. Following the evaluation strategy most often employed in previous studies including [50, 65, 66, 80], we also used the uncontrolled keyphrases for evaluation purposes (as only 18% of the controlled keyphrases were present in the abstracts, as opposed to more than 76% for the uncontrolled terms).

As Hasan and Ng [46] also pointed it out, different authors using the Inspec dataset calculated the recall of their systems differently, which makes the direct comparison of their performance scores problematic. The permissive evaluation – employed in [50, 65] for instance – requires only those gold standard phrases to be predicted by a system to achieve a perfect recall that can be found within the abstracts. A more restrictive evaluation, employed in [46, 80] for instance, does not take into consideration whether the gold standard keyphrases can be found in the abstracts; when a gold standard keyphrase is not returned by a system, it is counted as a false negative decision in all circumstances. In most cases, it is clear what kind of evaluations the authors of previous studies employed, as they either stated it explicitly, or it could be inferred from their results. There are some unfortunate cases, however, where it is not entirely clear which criterion the authors chose to report their results.

For the above-mentioned reasons, we regard our results based on the official evaluation script and relying on the standard benchmark dataset of the SemEval shared task more suitable for the comparison of the performances of different approaches. The other reason why we regard performing comparisons on the SemEval dataset more favorable is that it contains full documents as opposed to the Inspec dataset, which consists of scientific abstracts. This can be important, as several previous studies suggested [19, 46, 66] that systems performing well on the extraction of keyphrases from short documents often suffer from a substantial loss in performance when they need to extract keyphrases from long documents. Nevertheless, results on the Inspec dataset might provide interesting additional insights into the performance of our framework.

Method	P@5	R@5	F@5	P@10	R@10	F@10	P@15	R@15	F@15
Baseline (BL)	14.8	6.2	8.7	10.0	8.3	9.1	8.2	10.2	9.1
BL+MWE	18.0	7.5	10.6	14.3	11.9	13.0	10.9	13.5	12.1
BL+WikiCategory	23.2	9.6	13.6	18.5	15.4	16.8	15.7	19.6	17.5
BL+Orthography	28.0	11.6	16.4	21.3	17.7	19.3	16.9	21.1	18.8
BL+POS	26.6	11.1	15.6	22.7	18.9	20.6	18.8	23.4	20.9

Table 4.3: Results obtained by adding one extra feature to our baseline feature set at a time, evaluated against reader-assigned keyphrases of the SemEval dataset

Method	P@5	R@5	F@5	P@10	R@10	F@10	P@15	R@15	F@15
Baseline (BL)	19.2	6.6	9.8	13.4	9.1	10.9	10.9	11.1	11.0
BL+MWE	23.4	8.0	11.9	17.7	12.1	14.4	13.4	13.7	13.6
BL+WikiCategory	31.4	10.7	16.0	24.4	16.6	19.8	20.4	20.9	20.6
BL+Orthography	35.6	12.1	18.1	26.5	18.1	21.5	21.3	21.8	21.6
BL+POS	33.2	11.3	16.9	27.8	19.0	22.5	23.1	23.6	23.3

Table 4.4: Results obtained by adding one extra feature to our baseline feature set at a time, evaluated against combined keyphrases of the SemEval dataset

We should also add that although the appropriateness of both the permissive and the restrictive evaluations can be argued, we consider the latter kind of evaluation to be more appropriate, as this way only those systems that return all the keyphrases determined by a professional indexer (irrespective of whether the gold standard keyphrases are present in the documents) can be awarded with a perfect recall score. For this reason, we report our results using the restrictive evaluation schema and we also explicitly indicate in Table 4.8 the kind of evaluation that was employed by the authors of other papers.

4.3.2 Evaluation

Next, we detail the results obtained by our models on the two datesets introduced previously. Unless stated otherwise, results will be reported in the form of precision, recall and F-score values (abbreviated as P, R and F, respectively), as introduced in Section 1.1.3.

Evaluation on the SemEval dataset

We built our baseline model based on the feature set of KEA and added one feature at a time to learn their contribution to the overall performance. The effects of extending the baseline feature set with features described in Section 4.2.3 are illustrated in Table 4.3 and 4.4 for the evaluation on the SemEval dataset against the reader-assigned and combined gold standard keyphrases, respectively. Our models that used one additional kind of feature consistently beat all of the baseline performance scores with a large margin for all the evaluation scenarios. Nevertheless all of the proposed features proved their usefulness, it was important to see their joint effect on the overall performance.

Method	P@5	R@5	F@5	P@10	R@10	F@10	P@15	R@15	F@15
Merged	31.6	13.1	18.5	24.5	20.4	22.2	19.5	24.3	21.6
$\text{Merged}_{\text{BIES}}$	31.2	13.0	18.3	23.9	19.9	21.7	19.9	24.8	22.0
$\text{Merged}_{\text{BIES+CF}}$	32.4	13.5	19.0	23.8	19.8	21.6	20.0	24.9	22.2

Table 4.5: Effect of the use of the non-sequential features and candidate selection against the reader-assigned gold annotation on the SemEval dataset

Method	P@5	R@5	F@5	P@10	R@10	F@10	P@15	R@15	F@15
Merged	39.0	13.3	19.8	30.4	20.7	24.7	24.4	25.0	24.7
$\text{Merged}_{\text{BIES}}$	39.2	13.4	19.9	30.2	20.6	24.5	24.9	25.4	25.2
$\text{Merged}_{\text{BIES+CF}}$	40.6	13.9	20.7	30.2	20.6	24.5	24.9	25.4	25.2

Table 4.6: Effect of the use of the non-sequential features and candidate selection against the combined gold annotation on the SemEval dataset

The results of our classifier can be seen in the first lines of Table 4.5 and 4.6 when combining all the features into a single model and evaluating it on the SemEval dataset against reader-assigned and combined gold standard keyphrases, respectively. In these tables *Merged* refers to the fact that these models merged all the previously described features into a single model. Examining these results, we can see that the gains of the different views of the candidates were able to add up and produce an even better performance.

Merging all the feature templates resulted in a feature set that consisted of more than 30,000 elements. One of the reasons for this was the use of entire POS and named entity tag sequences as features and the other was the usage of the Wikipedia categories. Feature counts on that (and even much bigger) scales are not irregular in natural language processing tasks, but if we add that we had positive training instances on the scale of 2,000, the need for a reduction in the number of features can be argued.

In order to empirically test our hypothesis on data sparsity when using a rich feature set, we replaced features which encoded entire sequences of tags by a series of per token position-label pairs, as described in Section 4.2.3. The second rows (marked with the *BIES* subscript) in Table 4.5 and 4.6 list the results obtained when non-sequential tag features were applied instead of sequential ones. Using non-sequential features not only reduced the dimensionality of the feature space, but it also slightly improved the quality of the keyphrases which were returned as the best 15 phrases. As the main ranking criterion of the systems participating at the shared task was based on the performance of their top-15-ranked keyphrases, this kind of feature representation was employed in our subsequent experiments.

Next, the effects of the candidate filtering (*CF* for short), as described in Section 4.2.2, were examined. Candidate filtering lessened the effect of the overly dominant nature of the non-proper training instances. As a result of applying the proposed techniques, over 45% of the training instances were discarded (see Table 4.1), but the quality of the keyphrases extracted remained at the same

	Reader			Combined			Author		
Method	@5	@10	@15	@5	@10	@15	@5	@10	@15
HUMB	17.8	22.5	23.5	19.8	26.0	27.5	**23.9**	22.2	19.3
$\text{Merged}_{\text{BIES+CF}}$	**19.0**	21.6	22.2	**20.7**	24.5	25.2	**23.9**	20.0	16.6
WINGNUS	18.0	21.4	22.0	20.5	24.7	25.2	21.0	18.2	14.8
Maui	14.7	16.4	16.1	17.8	20.4	20.6	23.0	19.8	16.2

Table 4.7: F-scores achieved on the SemEval dataset by our final model and top-ranked shared task participants

level or even increased,as can be seen in the third rows of Table 4.5 and 4.6.

As regards comparative results with the performance of other shared task participants, our system performed as well as any of them when evaluated for the top-5 keyphrases, and it was only the system HUMB [67] – which used extra training data besides the corpus provided by the organizers – that achieved better performance scores against all the three (i.e. reader, combined and author) gold standard sets for the top-10 keyphrases (see Table 4.7). Our system ranked second – again behind HUMB – for evaluations against the top-15 keyphrases. The paper describing HUMB reports that their combined test set performance evaluated for the top-15 keyphrases improved by 7.4% due to the additional training data they used.

Looking at the results of WINGNUS and Maui systems, it is interesting to note that WINGNUS tends to perform better on evaluations against the reader-assigned keyphrases, while its relative performance degrades severely on evaluations against author keyphrases and the opposite holds for Maui. The performance of our system, however, seems to exhibit a more robust performance over different evaluation settings.

The official ranking of the shared task was based on the top-15-ranked keyphrases. However, as both the median and the mode of the number of gold standard keyphrases on the test set were below 15 – i.e. they were 11 and 3 for reader and author-assigned keyphrases, respectively – we think that evaluations performed at some lower threshold are more relevant when judging the utility of keyphrase extraction systems on this dataset.

The following example demonstrates the strictness of the evaluation applied in the shared task for one of the test set documents – entitled *Trading Networks with Price-Setting Agents* – as the official scorer returned a document-level F-score of value 0 for the predicted set of Porter-stemmed keyphrases, being *trader, nash equilibrium, game theori, price, network format, buyer, seller, market microstructur, agent, trade, posit profit, price-set agent, bid, mechan design, subgam perfect nash equilibrium.*

For the above document, the expected set of combined (i.e. reader or author) Porter-stemmed phrases were: *algorithm game theori, market, trade network, interact of buyer and seller, initi endow of monei, bid price, perfect competit, benefit, maximum and minimum amount, econom and financ, strateg behavior of trader, complementari slack, monopoli, trade network.*

Method	Evaluation	P	R	F
Hulth [50]	permissive	0.252	0.517	0.339
$\text{Merged}_{\text{BIES+CF}}$	restrictive	0.281	0.430	0.340
TextRank [80]	restrictive	0.312	0.431	0.362
KeyCluster [65]	permissive	0.350	0.660	0.457

Table 4.8: Results for previously published systems and our model on the Inspec dataset

Inspecting the title or the set of gold standard phrases of the document, the predicted keyphrases – in contrast to the document-level F-score they account for – are arguably not entirely useless.

Missed phrases in the gold standard set were often super-phrases of some predicted phrase or vice versa, e.g. *algorithm game theori* and *game theori* or *market* and *market microstructure*. There were also phrases in the gold standard set, the meaning of which could be composed from distinct elements of the predicted set, such as *bid price* versus *bid* and *price*.

Evaluation on the Inspec dataset

The results achieved by our method on the Inspec dataset along with the performance scores of previously published approaches on the same dataset can be found in Table 4.8 and 4.9. Table 4.8 also explicitly states what kind of calculation (i.e. permissive or restrictive) was employed in the previous studies during the calculation of their recall scores. It can be seen that our approach performs competitively with previously published results on that dataset. We should add that approaches on the Inspec dataset tend to achieve high results more easily, as abstracts are typically short, which results in a larger proportion of the candidate terms being useful than those on the scenario where keyphrases need to be extracted from complete documents. This assumption is in accordance with the observations of others, e.g. [19, 46]. Bougouin et al. [19] re-implemented the TextRank algorithm and evaluated its performance on the SemEval dataset against the combined set of author and reader-assigned keyphrases, which resulted in an F-score of 5.6. Out of the 19 participants of the shared task, 18 achieved better results than this.

Only the KeyCluster approach – based on the clustering of keyphrase candidates as described in [65] – seems to be superior to all other existing frameworks on the Inspec dataset. We should remark, however, that these results were obtained via the permissive calculation of the recall values. Obviously, if the authors reported their evaluation in the more restrictive manner, their results would be somewhat lower (yet better than other approaches, but with a much narrower margin) – as also pointed out by Hasan and Ng [46]. Another concern for the KeyCluster algorithm is that its authors reported their best performances, when they chose the number of clusters, m, as a function of the keyphrase candidates, n, as either $m = \frac{2}{3}n$ or $m = \frac{4}{5}n$ when performing hierarchical clustering and spectral clustering, respectively. This suggests that it might not be the clustering that is really beneficial in that approach, but the candidate generation step preceding it, as the best results were obtained when the number of clusters were not chosen to be considerably smaller compared to the

Method	P@5	R@5	F@5	Bpref	MRR
Topical PageRank [66]	0.354	0.183	0.242	0.274	0.583
$\text{Merged}_{\text{BIES+CF}}$	0.381	0.194	0.257	0.326	0.657

Table 4.9: Comparison of our results with those of the Topical PageRank approach on the Inspec dataset

extracted number of candidate terms. Due to the shortness of the documents in the Inspec dataset, i.e. 136.3 tokens per document on average as reported in [19], this approach could produce effective results. However, the performance of this approach is likely to degrade severely if it was evaluated on the SemEval dataset that consists of documents having an average length of 5179.6 tokens per document, as also reported in [19].

Indeed, the authors of KeyCluster algorithm reported in their other study [66] that the clustering-based method performed poorly on long articles (not originating from the SemEval dataset). In their paper, they claimed that the Topical PageRank (TPR) algorithm was better at handling longer documents as well. For this reason, we compared their results reported on the Inspec dataset with the performance of our system. The effectiveness of the TPR algorithm was characterized by two additional measures besides precision, recall and the F-score, namely the binary preference measure [23] and the mean reciprocal rank [115]. These measures not only take into consideration the proportion of the correctly determined keyphrases at some given threshold, but also account for the quality of the ranking of keyphrases. The binary preference measure (Bpref) measure can be calculated by the formula

$$Bpref = \frac{1}{R}\sum_{r} 1 - \frac{|n \text{ ranked higher than } r|}{R},$$

for a document with R relevant keyphrases, r being a relevant keyphrase of the document and n is a member of the first R nonrelevant keyphrases that were returned by a system. Mean-reciprocal rank (MRR) evaluates the quality of the extracted keyphrases by looking at the positions of the first correctly extracted keyphrases for each document in the test collection. MRR is calculated as follows

$$MRR = \frac{1}{|D|}\sum_{d \in D} \frac{1}{rank_d},$$

where D is the set of test documents and $rank_d$ denotes the rank of the first extracted keyphrase that is also present in the set of gold standard keyphrases for document $d \in D$. Our results and those of TPR can be found in Table 4.9, from which we can see that our approach consistently outperformed the TPR algorithm on all the evaluation metrics.

4.4 Related work

Here, we present the most common approaches used for keyphrase extraction and also give an overview on previous attempts to exploit extra-document information in keyphrase extraction. Methods applied by top-ranked shared task participants will also be introduced here for purposes of comparison.

Supervised and unsupervised solutions

GenEx [111] was one of the first systems to treat keyphrase extraction as a supervised learning task. It was a combination of the Genitor genetic algorithm and the module Extractor for extracting the keyphrases. Genitor was used in order to maximize the performance of Extractor by tuning the weights of 12 features that described keyphrase candidates. Hulth [50] pointed out in her study how incorporating linguistic knowledge could improve the performance of keyphrase extraction.

The statistics-driven approach in [109] used multiple language models to rank phrases based on their pointwise Kullback-Leibler divergence. This approach favored those phrases that received high probability values from a higher order in-domain (called the foreground) language model as opposed to some unigram out-of-domain (called the background) language model, which criterion resulted in that highly ranked phrases were of sufficient *phraseness* and *informativeness*. This approach was intended to overcome the shortcomings of the *binomial log-likelihood ratio test* employed by Dunning [34] in order to find frequent collocations in text. One possible drawback of such methods is, however, that the quality of the collocations identified can vary greatly due to different choices of the background language model.

Mihalcea and Tarau [80] introduced TextRank, which adapts the idea of the PageRank [90] algorithm to the extraction of important keyphrases and sentences from documents. This approach inspired many further studies, including [19, 66]. The authors of these papers introduced various unsupervised methods that incorporated the simulation of random walks performed on the co-occurrence graphs built from keyphrase candidates. The unsupervised framework presented in [65] was also based on co-occurrences, but its authors chose a clustering approach rather than a random walk-based one.

The Topical PageRank (TPR) approach [66] first determines a set of latent topics based on some document collection, then handles documents as mixtures of those topics relying on Latent Dirichlet Allocation (LDA) topic modeling [17]. Then the topic-aware rankings of candidate terms are composed upon the determination of the keyphrases of a document with a certain topic distribution.

These approaches tend to perform well on the extraction of keyphrases from short passage texts (e.g. from scientific abstracts), but, as reported in several previous studies [19, 46, 66], their performance scores severely degrade when they are utilized for the extraction of keyphrases from longer texts (e.g. full scientific papers). Our evaluation results presented in Section 4.3 suggest that our

proposed solution has the advantage of performing more consistently and competitively than other state-of-the-art systems, irrespective of the document length from which keyphrases are extracted.

Generation of keyphrase candidates

Existing systems can also be distinguished on the basis of the generation of keyphrase candidates. One way of carrying out candidate phrase extraction is the completely uncontrolled way, which means that essentially any successive tokens – except for those starting or ending with stopwords or punctuation – are treated as potential keyphrases, as was done in KEA [119]. Different candidate phrase generating strategies are summarized in the article by Kim and Kan [54], which covers both aspects of candidate selection and feature engineering for the extraction of keyphrases from scientific articles.

Other systems may require phrase candidates to satisfy certain requirements – e.g. to be part of a noun phrase, as was the case in the study by Barker and Cornacchia [5]. You et al. [124] used the so-called core word expansion algorithm, which first finds a set of core words and the final set of candidate phrases are generated from these seed phrases. They claimed that their method might reduce the candidate set by about 75%.

The KEA++ system [78], however, uses a controlled indexing strategy, meaning that candidate phrases are retrieved with the help of a domain-dependent thesaurus. The use of a thesaurus can be viewed as a way of incorporating extra-textual information into keyphrase extraction and its use prevents many ill-formed phrases from being handled as keyphrases. Having its advantages, this kind of approach may also exclude genuine keyphrases from the set of candidate terms and the availability of a topic-dependent thesauri is not necessarily the case for arbitrary domains. Domain dependent thesauri can be replaced by relying on Wikipedia instead, as was done in [118], but despite its wide coverage, the possibility of the exclusion of proper keyphrases cannot be ruled out.

Handling semantic relatedness

One of the key issues that need to be addressed in keyphrase extraction is that of recognizing semantic relatedness among terms. The classic approaches are based on an analysis of term-document co-occurrence, involving e.g. Latent Semantic Indexing [61] or metrics derived from the path between the concepts of some taxonomy, usually from the hypernym tree of Wordnets, as in [96]. The articles of [24, 92] contain a detailed description of WordNet-related semantic relatedness measures. These approaches, however, might suffer from the lack of desirable coverage of the taxonomies that they employ.

Extra-textual information to augment the consistency of phrases extracted from documents was used in the form of Web queries by Turney [113]. Some other articles like [69] incorporate citations-derived information into their models to improve their results.

Wikipedia was widely used earlier in tasks that attempted to determine semantic relatedness among concepts, e.g. [41, 85, 102, 123]. Our proposed approach is related to these earlier studies by the fact that we employ the category hierarchy of Wikipedia concepts as an external source of information to enhance the quality of keyphrase extraction.

The approach applied in this thesis belongs to those supervised keyphrase extraction approaches, which use various ways (without relying on any thesaurus) to control the set of keyphrase candidates. Semantic knowledge was also incorporated into our system by relying on both Wikipedia and WordNet. We showed – by exploiting its category structure – that the semantic knowledge incorporated in Wikipedia can be utilized without the need to build massive indices on all its textual contents.

Description of top-performing and related participant systems

As we also participated in the SemEval shared task, here we will introduce some of the participating team's approaches which were either top-performing or had some relatedness to our framework. A complete description of the 19 participating systems can be found in the overview paper of the shared task [56]. To the best of our knowledge, the results achieved by the best-performing shared task participants still act as an upper-bound for the performance scores reported in more recent articles which apply the same dataset and scoring strategy of the shared task, like [19, 121]. For this reason, comparative results will be reported with respect systems that participated in the shared task.

The approach employed in Maui [79] arguably lies the closest to ours as it retrieves various metrics from Wikipedia to use them as features describing the keyphraseness of candidate phrases. One clear distinction between the two approaches is that while Maui uses all the textual contents of Wikipedia (e.g. upon calculating the probability of finding some keyphrase candidate as an anchor text), we only rely on its category hierarchy. The approach applied by Maui needs a massive index of textual occurrences from a Wikipedia dump – something that our approach does not require, still being able to outperform it.

Some of the top-ranked shared task participants, like HUMB [67] and WINGNUS [89] crawled the original PDF articles and processed them – instead of relying on the plain text versions provided by the organizers – which gave them the chance to examine the logical structure of documents more precisely. Despite the fact that our approach did not enjoy such benefits, it performed competitively with these systems and a possible line of future research might be to use a combination of semantical and structural features derived from documents.

HUMB not only used Wikipedia as an external resource, but also GRISP [68] – being a large-scale terminological database derived from multiple resources – as a mean for discriminating between proper and improper keyphrases. Some of the Wikipedia-based features that were originally introduced in Maui were employed in that work as well.

One of the key problems which have to be addressed in keyphrase extraction is that of recognizing semantic relatedness between potential phrases and the documents in which they occur. The

classic approaches for this are co-occurrence-based measures, e.g. Latent Semantic Indexing-based approaches such as that of [61] and metrics derived from the path between the concepts of some taxonomy, usually from the hypernym tree of Wordnets, like in [96].

Relatedness calculated from co-occurrences may be noisy, while the coverage of taxonomies is generally low. To overcome these disadvantages, studies have suggested the usage of the semi-structured Wikipedia as the source of semantic relatedness information. WikiRelate! [102] used the redirect and disambiguation pages and the category hierarchy of Wikipedia.

4.5 Summary of thesis results

In this chapter, the author described his contributions to the task of keyphrase extraction from scientific documents. Related to his publications [7, 8, 84], the author regards the following as his main contributions to the research topic:

1. Extending the existing keyphrase candidate filtering techniques Novel ways of preventing false positive keyphrase candidates from becoming classification instances were introduced based on the utilization of WordNet and lexical patterns. By ruling out keyphrase candidates which are unlikely to act as a keyphrases, the underrepresented nature of keyphrases during training can be reduced. Being able to prevent certain phrases from becoming keyphrase candidates is also beneficial when making predictions, as certain false positive keyphrase candidates can be guaranteed not to be erroneously predicted as keyphrases.

2. Introducing condensed representations for sequential features An alternative way for representing features encoding sequences of observations was proposed. The new features had the advantage of being able to encode a similar amount of useful information that would otherwise have been stored in sequential feature, but using a smaller number of them.

3. Utilizing of extra-textual information for representing keyphrase candidates Features involving extra-textual information were derived from Wikipedia in multiple ways that do not require all the enormously big textual content of Wikipedia to be indexed, as the proposed features are purely derived from the links and other formatted textual parts of Wikipedia. Wikipedia was used for the generation of MWE-related features and as a way of describing keyphrase candidates by their semantic categories retrieved from the category structure of Wikipedia.

While Wikipedia-derived features gained significant improvements over baseline solutions, the filtering of keyphrase candidates and the condensed representation of sequential features made it possible to reduce the number of training instances and model features, thus avoiding problems related to data sparsity and the curse of dimensionality.

The effectiveness of the contributions to the task was shown on the dataset of the shared task of SemEval-2, which aimed the extraction of keyphrases from scientific documents. The model proposed in this thesis always outperforms or closely matches the results that have been previously reported under various evaluation criteria on this dataset in other studies. SZTERGAK [8], being the author's predecessor framework actually, which participated in the SemEval-2 shared task [55], was ranked 3$^{\text{rd}}$ out of 19 participating teams in the official evaluation campaign carried out by the shared task organizers.

Chapter 5

Opinion Phrase Extraction

There has been a constant growth in the amount of user-generated on-line contents, including those published on reviews sites like `epinions.com`. Monitoring these kinds of sources has became a valuable source of information for companies, which might help them to plan their marketing strategies. Analyses of the utterances of users on blogs and review sites – where they usually express their opinions about various issues – can clearly serve as the basis of marketing strategies, but their enormous size calls for the need of automatic approaches. The goal of opinion mining is to identify users' opinions towards some target entity, based on some textual data that they wrote. In this chapter, we introduce a keyphrase extraction-based approach for the extraction of key argument phrases which make the opinion holder feel negative or positive towards a particular product.

5.1 Motivation

There has been a growing interest in the NLP treatment of subjectivity and sentiment analysis, see e.g. [3, 64, 91, 108], and the task of keyphrase extraction – as described in previous chapters – has also received significant academic interest. Product reviews serve as perfect targets for the combination of the above-mentioned research areas as the *opinion phrases* of product reviews can be interpreted analogously to "regular" keyphrases of textual documents, i.e. they are those phrases which play a decisive role within the document they are included. The fact that some review portals allow users to leave a set of pro and con phrases – just like scientific or newswire publications are often accompanied by sets of keyphrases – underlines the resemblance between opinion phrases and scientific keyphrases. However, pros and cons given by reviewers are often inappropriate (e.g. "none" for pros and "everything" for cons) or totally absent. These phrases are neither exclusive most of the times, i.e. some important aspects are mentioned in the review that are not covered in the pros and cons part of a review or vice versa. Owing to these phenomena, the automatic generation of opinion phrases can be a useful way of analyzing product and service reviews.

Despite the somewhat common nature of opinion phrases of reviews and keyphrases of scientific articles, methods that work on the well-studied field of scientific keyphrase extraction are not necessarily directly applicable successfully to the extraction of opinion phrases from product reviews. On the one hand, although proper phrases have their decisive role in both types of genres, opinion phrases are those that form the sentiments of the opinion holder, whereas in the case of scientific keyphrases they should be such phrases that summarize well the content of a document. Note the difference between opinion phrases and those which summarize well the contents of a document, i.e. one can frequently use such phrases in a review that does not have much importance in the opinion-forming aspect, whereas in the case of scientific documents frequently used phrases tend more to be proper keyphrases. For these reasons, when designing our model, we paid special attention on incorporating such features, bridging the gap which arise from the domain differences of scientific literature and product reviewing.

5.2 Keyphrase Extraction Framework

The basic principles of the framework employed in this chapter resembles those described in Chapter 4, which means that in order to extract opinion phrases from reviews, we used supervised machine learning techniques. Candidate terms were similarly extracted from the reviews and those present in the gold standard set of pros and cons were treated as positive examples during the training and evaluation phases. Maximum Entropy classifiers were trained with different feature sets and the keyphrase candidates with the highest a posteriori probabilities were selected as keyphrases for the product reviews in the test collection.

5.2.1 Generation of keyphrase candidates

As also mentioned in Chapter 4, filtering of the candidate phrases retrieved from documents can be a useful preprocessing step in keyphrase extraction. This may be beneficial as the number of n-grams that might be extracted and that of genuine keyphrases out of them is often highly imbalanced in documents.

Most of the criteria for treating a successive n-gram as a potential keyphrase were the same as those described earlier in Section 4.2.1, i.e. the length and the extent to which stopwords were included among the occurrences of a normalized keyphrase candidate were taken into account. The only filtering criterion not employed here, which was discussed in Section 4.2.1 imposed restrictions based on the within-document position of candidate phrases, i.e. keyphrase candidates *only* being present among the references of a publication were discarded. The reason for not applying any similar restriction when keyphrase candidates were extracted from product reviews is due to the fact that the structure of product reviews is more flexible compared to that of scientific publications, resulting in the fact that sensible restrictions that hold generally enough for the within-document

positions of opinion phrase candidates cannot be drawn easily.

As a consequence, normalization of the candidate phrases was performed, which was intended to bring orthographically different, yet semantically equivalent occurrences to some common form. The normalization of the candidate phrases was carried out in an analogous way as it was described in Section 4.2.2, which included the lowercasing, Porter-stemming of the lemmas of the constituents of a candidate phrase, followed by the alphabetical reordering of the remaining tokens and the omission of stopwords. With this kind of representation it was then possible to handle two phrases, such as '*screen is tiny*' and '*TINY screen*' in the same way.

Experiments involving the extra normalization step, based on the utilization of WordNet – in the very same manner as introduced in Section 4.2.2 – was employed during the extraction of opinion phrases as well. This made it possible to handle two originally differently stemmed word forms, such as *decide* and *decision* to be handled by the same normalized form.

Next, candidate terms were handled at the review level instead of the occurrence level. This meant that the normalized forms of keyphrase candidates were classified based on their overall behavior over a review, instead of classifying their individual occurrences within the review.

5.2.2 Feature representation

Our feature set was constructed to represent review-level keyphrase candidates. The feature space incorporates features calculated on the basis of the normalized phrases themselves, but more importantly, owing to the mapping between the normalized phrase forms and their original occurrences, it was possible to add new contextual orthographic and syntactic features as well.

Next, we will describe the features applied for characterizing opinion phrase candidates. For the sake of completeness, the full set of features – some elements of which overlap with those introduced in Section 4.2.3 – will be described here.

The features also presented in Section 4.2.3 are briefly introduced and illustrated by some domain-specific examples. More importantly, we shall focus on those task-specific features (such as those which rely on SentiWordNet) that were specially designed for the novel task of opinion phrase extraction. To make the non-redundant features clearly distinguishable from those being introduced earlier, those features that were uniquely applied for the task of opinion phrase extraction are marked by asterisks.

Standard Features

Since we assumed that the underlying principles of extracting opinionated phrases were analogous to that of extracting standard (mostly scientific) keyphrases, features of the standard setting were applied in this task as well. The most common ones, introduced by KEA [119] are the **Tf-idf** value and the **relative position** of the first occurrence of a candidate phrase within a document. **Phrase**

length is also a common feature, which was defined here as the number of non-stopword tokens an opinion phrase candidate consisted of.

Linguistic and orthographic features

POS tags-derived features Since certain POS-tags tend to be more frequent than others for genuine keyphrases, features generated by POS-tags belonging to the occurrences of a normalized phrase were applied. To overcome the issue of introducing a new feature for every possible POS-tag sequence, the solution proposed earlier in Section 4.2.3 was employed here as well, i.e. the *S-*, *B-*, *E-* and *I-* prefixes were used to indicate the within-phrase position of POS-tags. For instance, the phrase *cheap/JJ phone/NN* made the features *B-JJ* and *E-NN* to fire, however, the one-token phrase *cheap/JJ* resulted in the activation of the single feature *S-JJ*.

Orthography-related features Opinionated phrases often behave specially from an orthographic perspective, e.g. in the case of *so sl**ooo**w* or *CHEAP*. Owing to the fact that the running text forms of the normalized phrase candidates were stored in our representation, it was possible to construct features for these phenomena. One of them is responsible for **character run**s* (i.e. more than 2 of the same consecutive characters), and another is responsible for **strange capitalization** (i.e. the presence of uppercase characters besides the initial one). The identification of **Named Entities** (NEs) is closely related to the orthographic behavior of the tokens, which was also taken into consideration during the feature representation. Features indicated the presence of an NE (with its standard 4-way CoNLL-style category – i.e. person, location, organization or miscellaneous – as well) for opinion phrase candidates. The *S-*,*B-*,*E-*,*I-* within-phrase locational clues were incorporated into these features as well.

Character suffices* Features generated from the **character suffixes** of the individual tokens of the occurrences of a normalized keyphrase candidate were also employed. Character suffix features also incorporated positional information, similar to POS features. The suffixes themselves came from the last 2 and 3 characters of the tokens constructing an n-gram. For instance, the features induced by (and thus assigned with true value) for the phrase *ch**eap** ph**one*** are {*B-eap*, *B-ap*, *E-one*, *E-ne*}.

Syntax-dependent features* We introduced features that exploited the syntactic context of a candidate with parse trees. For an n-gram with respect to all the sentences it was contained in a given document, this feature stored the average and the minimum depths of those **NP-rooted trees** that contained the whole n-gram as its yield. These features are intended to express the "noun phraseness" of the phrase.

World knowledge-based features

Features relying on the external resources of Wikipedia and SentiWordNet were also exploited by our model, which proved to be useful as world knowledge could be incorporated by relying on them.

Wikipedia-based features Wikipedia was used to incorporate semantic features from its category hierarchy in a similar way as it was introduced in Section 4.2.3. In the case of a candidate phrase, all the nominal parts of the normalized titles of **Wikipedia categories** for its related Wikipedia articles were added as separate binary features to the feature space. The normalization of the Wikipedia category names was performed in the same way as that of keyphrase candidates. For instance, given the candidate phrase '*service quality*' the feature *wiki_control_qual* was set to true as the Wikipedia article entitled *Service quality* was assigned to the Wikipedia category, named *Quality control*.

Multiword expressions Features related to the presence of **multiword expressions** for the opinion phrase candidates were employed in a similar way to that described in Section 4.2.3.

To demonstrate the added value of MWEs in the task of opinion phrase extraction, binary features were introduced here as well to indicate whether a certain n-gram (1) was an MWE in its full length, (2) could be built up from more MWEs, or just simply (3) was the superstring of at least one MWE from the list. Example candidates for these categories include phrases *ease of use*, *mobile internet access* (which can be assembled from phrases *mobile internet* and *internet access*) and *send text messages* (due to the presence of *text message* on our list of expressions).

SentiWordNet-based features* A more sophisticated surface-based feature used external information as well on the individual tokens of a phrase. It relied on the **sentiment scores** of SentiWordNet [2], a publicly available database that contains a subset of the synsets of the Princeton WordNet with positivity, negativity and neutrality scores assigned to each one, depending on the use of its sentiment orientation (which can be regarded as the probability of a phrase belonging to a synset being mentioned in a positive, negative or neutral context). These scores were utilized for the calculation of the sentiment orientations of each token of a keyphrase candidate. Surface-based features calculated on the basis of SentiWordNet measured the *maximal positivity and negativity and subjectivity* scores of the individual tokens and the *total sum* over all the tokens of a keyphrase candidate.

Sentence-based features were also defined based on SentiWordNet as it was also used to check for the presence of **indicator terms** within the sentences containing a candidate phrase. Those word forms were gathered from SentiWordNet for which the sum of the average positivity and negativity sentiments scores among all its synsets were above 0.5 (i.e. the ones that are more likely to have some kind of polarity). Binary features were then assigned to a candidate phrase reflecting which

polarity-indicating words did they co-occurred with.

SentiWordNet was also used to define features that describe keyphrase candidates from the perspective of the whole document, not just their local context. A sentiment score was assigned to each sentence – derived from the sentiment scores of the individual tokens comprising them. Then the mean and the deviation of these per-sentence sentiment scores were calculated and assigned to opinion phrase candidates. The mean of the sentiment scores yielded a general score on the **sentiment orientation** of the sentences containing a candidate phrase, while higher values for the **deviation** was intended to capture cases where a reviewer writes both factual (i.e. uses few opinionated words) and non-factual (i.e. uses more emotional phrases and opinions) sentences about a product.

Document and corpus-level features

Among document-level features, the **standard deviation of the positions** (normalized by the document length) was computed. Higher values of the standard deviation in the position means that the reviewer repeats some phrase at different points of the review, which might indicate that this phrase is of greater importance for them.

As verbs are often decisive in expressing the sentiment polarity towards the noun phrases they accompany (e.g. "*I adore its fancy screen.*" versus "*I bought this phone one year ago.*"), a set of features was applied to assess **indicator verbs***. Features were introduced for verbs co-occurring with candidate expressions at least 100 times in the training corpus. For each verb-candidate phrase pair, the minimum height of the constituency subtree, which had both the verb and the candidate phrase in its yield (the verb preceding the candidate phrase) was determined. The feature value assigned to features representing these verb-candidate phrase relations were simply taken as the reciprocal of this syntactic distance. This way, the feature value was scaled between 0 and 1. (Note that for those verbs which never co-occurred with some candidate phrase in any constituency tree, the syntactic distance value was defined to be infinity, the limit value of the reciprocal of which is 0.)

Associating such a value with a candidate can be helpful, especially when we are trying to extract opinion phrases from the same domain to the training corpus, since the same kind of products tend to have similar and frequently used argument phrases to express sentiments.

5.3 Experiments and discussion

To see the effectiveness of the proposed features on different domains, datasets consisting of reviews of mobile phones and movies were created. We used standard keyphrase extraction evaluation metrics and baselines to evaluate our pros and cons extractor system.

	Mobiles	Movies
Number of reviews	2009	1962
Average sentence/review	31.9	29.8
Average tokens/sentence	16.1	17.0
Average keyphrases/review	4.7	3.2
Average keyphrase candidates/review	130.38	135.89
Median of sentence number	20	20
Median of token number	14	15
Minimum sentences per review	6	5
Minimum tokens per sentence	1	1
Maximum sentences per review	319	372
Maximum tokens per sentence	199	127

Table 5.1: Statistics on the size of the corpora

5.3.1 Dataset

We crawled reviews from `epinions.com` on two different product domains, namely mobile phones and movies. For both domains, 2,000 reviews were collected along with 50 and 75 reviews to measure inter-annotator agreement. The dataset is available at `http://rgai.inf.u-szeged.hu/proCon`.

Due to the fact that reviewers are often non-native English speakers, this corpus is quite noisy (similar to other user-generated contents) as run-on sentences and improper grammar are frequently used. The list of pros and cons that reviewers assigned to their posts was also inconsistent in the sense that some reviewers used full sentences to express their opinions in the pro and con section of their review, which was used by others mostly to provide a list of few token-long phrases, which served as a summary of their review (such as *cheap price* or *clumsy buttons*). Even if some user entered a list, the segmentation of its elements was marked in various ways among reviews (e.g. comma, semicolon, ampersand or the *and* token) and even differed sometimes within the very same review. There were many general or uninformative pros and cons (like *none* or *everything* as a pro phrase) as well.

In order to have a consistent gold-standard annotation for training and evaluation, we refined the pros and cons of the reviews in the corpora. An automatic segmentation method split the list of pros and cons along the most frequent separator characters. This kind of segmentation of pro and con lists into pro and con phrases was then checked by two human annotators, who made corrections in 7.5% of the reviews. Then the human annotators also marked the general pros and cons (11.1% of the pro and con phrases) and the reviews which were left without any meaningful keyphrase were discarded.

Next, the filtered pros and cons were refined to be tag-like keyphrases. For instance, instead of the clause "*even I found the phones menus to be confusing*", we would like to have the phrase "*confusing phone menu*". Refinement was carried out both manually and semi-automatically, relying on hand-crafted transformation rules. These hand-crafted rules relied on a grammatical analysis (i.e. POS tags and parse trees) of the original pro and con text fragments, and proved to be domain

	Mobiles	Movies
Annotator-Annotator	85.7	78.0
Annotator-Automated	59.3	48.6

Table 5.2: Inter-annotator agreements expressed as F-score

independent (i.e. the same transformations were applicable for the mobile phone and the movie keyphrases). The manual refinement was carried out by two annotators and their inter-annotator agreement was measured using 50 mobile phones reviews and 75 movie reviews. In the two datasets 41.5% and 53.7% of the segmented keyphrases were modified by the transformation rule-based and the manual refinement procedures, respectively.

In order to investigate inter-annotator agreement, one of the annotator's decisions were regarded as correct and the F-score of the other annotator's decisions was calculated for the reviews refined by both of the annotators. As for the evaluation of the automatic phrase refinement procedure, we followed the same strategy, except that there it was possible to perform the previous calculations over the whole dataset (not just based on those overlapping reviews which were refined by both of the annotators). Table 5.1 shows the basic statistics concerning the size of the resulting corpora after the manual refinement steps and Table 5.2 lists the agreement rates on the refinement of the phrases of author-entered pros and cons.

From the inter-annotator agreement results, it can be seen that the domain of movies was more difficult for annotators as well. However, substantial agreement was reached among human annotators and the automatic procedure.

5.3.2 Evaluation

We conducted several experiments to learn the effectiveness of the proposed model under various circumstances. Firstly, we performed the standard automatic evaluation of the predicted sets of keyphrases, based on strict string matching to the set of gold standard opinion phrases. Secondly, a set of experiments was conducted involving human evaluation, as our error analysis suggested that the automatic evaluation – not being able to handle semantic similarities in certain cases – might severely underestimate the real usefulness of the predicted phrases. Next, experiments assuming the lack of abundant amount of training data were conducted, in the from of performing domain adaptation experiments.

Automatic evaluation

During our automatic evaluation a predicted keyphrase was only accepted as a true positive one if it exactly matched the standardized form of one of the gold standard keyphrases. The results reported here were obtained by performing 5-fold cross validation over the domains.

Method	P@5	R@5	F@5	P@10	R@10	F@10	P@15	R@15	F@15
KEA	1.7	1.8	1.8	1.4	3.0	1.9	1.4	4.5	2.1
Baseline (BL)	2.6	2.8	2.7	2.6	5.5	3.5	2.6	8.2	3.9
$\text{Baseline}_{\text{WN}}$	2.7	2.9	2.8	2.7	5.8	3.7	2.7	8.7	4.1

Table 5.3: Baseline results using a strict evaluation on the domain of mobile phones

Method	P@5	R@5	F@5	P@10	R@10	F@10	P@15	R@15	F@15
BL_{WN}	2.7	2.9	2.8	2.7	5.8	3.7	2.7	8.7	4.1
BL_{WN}+Indicator Verbs	3.1	3.4	$3.3^{\S}$	2.9	6.2	3.9	2.8	9.1	4.3
BL_{WN}+Length	3.2	3.4	$3.3^{\S}$	3.1	6.6	$4.2^{\dagger}$	2.9	9.3	4.4
BL_{WN}+MWE	4.7	5.0	$4.9^{\ddagger}$	3.8	8.0	$5.1^{\ddagger}$	3.4	10.8	$5.1^{\ddagger}$
BL_{WN}+POS	4.6	4.9	$4.7^{\ddagger}$	4.2	9.0	$5.8^{\ddagger}$	3.9	12.6	$6.0^{\ddagger}$
BL_{WN}+SentiWordNet	6.0	6.4	$6.2^{\ddagger}$	4.9	10.4	$6.7^{\ddagger}$	4.3	13.6	$6.5^{\ddagger}$
BL_{WN}+Standard Dev.	3.9	4.2	$4.1^{\ddagger}$	3.8	8.1	$5.2^{\ddagger}$	3.5	11.2	$5.3^{\ddagger}$
BL_{WN}+Orthography	3.2	3.4	$3.3^{\S}$	3.1	6.7	$4.3^{\dagger}$	2.9	9.5	4.5
BL_{WN}+Suffix	11.5	12.2	$11.8^{\ddagger}$	8.6	18.2	$11.7^{\ddagger}$	6.9	22.0	$10.5^{\ddagger}$
BL_{WN}+Syntax	3.5	3.7	$3.6^{\ddagger}$	3.0	6.4	4.1	2.8	9.1	4.3
BL_{WN}+WikiCategory	11.9	12.7	$12.3^{\ddagger}$	8.1	17.4	$11.1^{\ddagger}$	6.3	20.1	$9.6^{\ddagger}$
Merged	**14.8**	**15.7**	$\mathbf{15.3}^{\ddagger}$	10.4	22.0	$14.1^{\ddagger}$	8.0	25.4	$12.2^{\ddagger}$

Table 5.4: Results of the strict evaluation on the domain of mobile phones. Symbols §, † and ‡ indicate a significant improvement on the BL_{WN} system at confidence levels of 0.1, 0.05 and 0.01, respectively

As we treated the mining of pros and cons as a supervised keyphrase extraction task, we made the KEA framework [119] our baseline system, as this is one of the most cited, publicly available automatic keyphrase extraction systems. However, we should mention that because opinion candidates were defined in a slightly different way by our system and KEA, the two architectures might operate on different classification instances, for which reason alternative baselines were provided.

We defined a simple baseline solution, referred to as *Baseline* (BL for short) which followed the exact same strategy for identifying candidate phrases as our system did, but used the standard features of KEA. These features are the tf-idf scores and the relative first occurrences of keyphrase candidates. This alternative baseline provided a mean for validating the usefulness of the proposed approach for candidate phrase extraction, as it differed from KEA only in its strategy for defining candidate phrases.

Another baseline system called $Baseline_{WN}$ (BL_{WN} for short) was created to see the added value of the WordNet-based candidate normalization, without incorporating any novel features into our model. Thus, the only difference among the latter two baseline systems was that the previous baseline did not apply the WordNet-based normalization of phrase candidates, whereas $Baseline_{WN}$ did it so.

Comparing the performance scores in Table 5.3 and 5.5 validates the choices of both the alternative extraction of candidate phrases from documents (as opposed to KEA) and the employment

Method	P@5	R@5	F@5	P@10	R@10	F@10	P@15	R@15	F@15
KEA	1.2	1.9	1.5	1.0	3.1	1.5	0.9	4.3	1.5
Baseline (BL)	1.6	2.5	2.0	1.5	4.9	2.3	1.6	7.4	2.6
BL_{WN}	1.7	2.8	2.1	1.7	5.4	2.6	1.7	8.2	2.9

Table 5.5: Baseline results using a strict evaluation on the domain of movies

Method	P@5	R@5	F@5	P@10	R@10	F@10	P@15	R@15	F@15
BL_{WN}	1.7	2.8	2.1	1.7	5.4	2.6	1.7	8.2	2.9
BL_{WN}+Indicator Verbs	2.4	3.7	$2.9^{\dagger}$	2.0	6.3	$3.0^{\S}$	1.9	8.8	3.1
BL_{WN}+Length	2.1	3.3	2.6	2.0	6.4	$3.1^{\S}$	2.0	9.1	$3.2^{\S}$
BL_{WN}+MWE	2.3	3.6	$2.8^{\dagger}$	2.0	6.3	$3.1^{\dagger}$	1.9	9.1	$3.2^{\S}$
BL_{WN}+POS	2.9	4.6	$3.6^{\ddagger}$	2.8	8.7	$4.2^{\ddagger}$	2.5	11.7	$4.1^{\ddagger}$
BL_{WN}+SentiWordNet	3.7	6.0	$4.6^{\ddagger}$	3.1	9.8	$4.7^{\ddagger}$	2.8	13.1	$4.6^{\ddagger}$
BL_{WN}+Standard Dev.	2.9	4.6	$3.6^{\ddagger}$	2.6	8.1	$3.9^{\ddagger}$	2.5	11.6	$4.1^{\ddagger}$
BL_{WN}+Orthography	3.0	4.7	$3.7^{\ddagger}$	2.5	7.8	$3.8^{\ddagger}$	2.3	10.9	$3.8^{\ddagger}$
BL_{WN}+Suffix	6.8	10.7	$8.3^{\ddagger}$	5.2	16.4	$7.9^{\ddagger}$	4.3	20.1	$7.1^{\ddagger}$
BL_{WN}+Syntax	2.3	3.6	$2.8^{\dagger}$	2.0	6.1	$3.0^{\S}$	1.9	9.1	$3.2^{\S}$
BL_{WN}+WikiCategory	8.8	13.9	$10.8^{\ddagger}$	6.3	19.8	$9.6^{\ddagger}$	4.8	22.5	$7.9^{\ddagger}$
Merged	**10.0**	**15.8**	$\mathbf{12.2}^{\ddagger}$	7.0	21.9	$10.6^{\ddagger}$	5.3	24.6	$8.7^{\ddagger}$

Table 5.6: Results of the strict evaluation on the domain of movies. Symbols §, † and ‡ indicate a significant improvement on the BL_{WN} system at confidence levels of 0.1, 0.05 and 0.01, respectively

of WordNet during candidate phrase normalization. For this reason, all our models that we experimented with subsequently were an extension of the baseline approach $Baseline_{WN}$, i.e. the system which uses both the candidate set filtering and the WordNet-based normalization steps.

Next, a set of experiments was carried out to see the added values of the individual features. In order to do so, one feature was added at a time to our $Baseline_{WN}$ model. As we can see from Table 5.4 and 5.6, all the features were highly effective in the sense that expanding the baseline feature set by them improved our results. Moreover, as indicated in the tables, these improvements were statistically significant in the majority of the cases.

The fact that the highest F-scores for keyphrases were achieved when the number of extracted phrases was close to the average number of pro and con phrases per reviews (i.e. between 4.7 and 3.2 for mobiles and movies, respectively) suggests that our ordering of keyphrase candidates is quite effective (as once we find the number of keyphrases a document has, performance cannot really increase any more).

Human evaluation

Our error analysis revealed that false positive predictions were often near misses; that is, they overlapped (e.g. *ring tones* versus *included ring tones*) or were synonymous (e.g. *small keys* versus *tiny keys*) with some phrases in the gold standard keyphrases. The abundance of such cases suggested

	Author	*Annotator*$_1$	*Annotator*$_2$	*Annotator*$_3$
Author	–	0.415	0.324	0.396
Annotator$_1$	0.601	–	0.679	0.708
Annotator$_2$	0.452	0.696	–	0.713
Annotator$_3$	0.525	0.690	0.685	–

Table 5.7: Inter-annotator agreement among the author's and annotators' sets of opinion phrases. Elements above and below the main diagonal refer to the agreement rates expressed in Dice coefficient for pro and con phrases, respectively

that a fully automatic evaluation scheme – often applied for the evaluation of scientific keyphrase extraction tasks – can severely underestimate the real usefulness of opinion phrase extraction models (see Section 1.1.3).

As for an illustration, our framework – returning phrases including *size, small, design, stylish, display, keyboard, battery feature, function, signal* for a particular review, the gold standard annotation phrases of which were ***accumulator****, sensitivity,* ***compact****, sound, menu, language,* ***lightweight*** – was not able to capture any true positive phrases when automatic evaluation was applied. However, the underlined phrases in the response set of our system can definitely be aligned to the bold phrases of the gold-standard set, implying at partial credit to them.

In order to have more reliable performance scores, we decided to perform a human evaluation. We selected a 25-element subset of reviews from our dataset (related to the mobile phone Nokia 6610). Since we had the same findings as [20], i.e. that authors often omit several opinion forming aspects from their pros and cons listings that they later include in their review, we decided to determine alternative sets of pros and cons for these reviews.

Three linguists were hired to assign pro and con opinion phrases after reading the selected reviews. In order not to be biased by the opinion phrases determined by the author of a review, the annotators were only given the free-text part of the review, i.e. the original *Pros and cons* and *Bottomline* sections were removed. In this way, three different pro and con annotations were produced for each review. Besides the opinion phrases determined by the readers of the reviews, the pro and con phrases written by the original author were also at hand. This kind of parallel availability of author-assigned and reader-determined keyphrases was also present during the evaluation of scientific keyphrases, making different evaluation criteria applicable.

The extent to which the sets of author-assigned and reader-defined pro and con phrases overlapped are reported in terms of Dice coefficients in Table 5.7. Based on the alternative sets of opinion phrases, three different evaluation scenarios were performed to see the effectiveness of the final combined model on the 25-review subset of the dataset.

Firstly, annotators compared the automatically extracted phrases with the union of the alternative sets of opinion phrases. Taking the union of sets obviously causes the cardinality of the set of gold standard keyphrases to increase. With the bigger number of gold standard keywords, it is more

Method	P@5	R@5	F@5	P@10	R@10	F@10	P@15	R@15	F@15
$\cup$	72.8	20.6	32.1	66.8	33.5	44.7	63.5	46.9	53.9
$\cap$	46.4	27.8	34.8	41.6	44.9	43.2	37.1	56.7	44.8
Author	34.4	22.3	27.1	31.6	35.4	33.4	28.8	45.1	35.2

Table 5.8: Results of the **human evaluation**. $\cup$, $\cap$ and Author means when the automatic keyphrases were matched against the union, intersection of the keyphrases of three independent annotators and the keyphrases of the original author, respectively

probable that predicted keywords occur among them. At the same time, having a larger set of gold standard tags might affect the recall in a negative way, as more gold standard keyphrases need to be extracted.

Secondly, annotators compared the predicted phrases against the intersection of the alternative sets of opinion phrases. This approach is naturally expected to behave in the opposite way – with respect precision and recall – to the evaluation relying on the union of the alternative sets of opinion phrases. This setting measures the extent to which the most important aspects, i.e. those that are present in every alternative sets of keyphrases can be extracted from reviews.

Lastly, annotators had to decide on the correctness of the predicted opinion phrases by comparing them with the original sets of pros and cons, written by the authors of the reviews. Performance scores achieved when applying the above described three evaluation criteria are presented in Table 5.8. These results confirm our assumptions on the precision and recall scores for the different evaluation criteria and more importantly, the evaluation involving human inspection suggests that the automatic evaluation indeed severely underestimated the real usefulness of the predicted opinion phrases.

Opinion phrase extraction versus scientific keyphrase extraction

Comparing the task of extracting keyphrases from scientific publications – discussed in Chapter 4 – with that of product reviews, we can make two remarks. Firstly, the keyphrases of scientific documents are more universal, i.e. once know that some expression, such as *distributed computing*, is a good keyphrase for one scientific document, we can be more confident about it being a proper keyphrase for other publications within the same domain. However, in the case of reviews, phrases such as *pink color* can be easily mentioned in either opinionated and non-opinionated contexts. Secondly, besides scientific keyphrases being more *universal*, they are more *deterministic* in the sense that there are fewer ways to express good keyphrases, e.g. supposing *simulated annealing* is a proper keyphrase for a scientific document, it is unlikely that an automatic system would extract *imitated annealing*, whereas in the case of product review the gold standard keyphrases often differ from their mention in the text (e.g. *tiny keys* and *small buttons*).

Besides these differences, another source of false positive test cases during our evaluations was due to the incompleteness of the opinion aspects entered by the authors, i.e. not all the important

aspects were necessarily listed among the pros and cons of a review, as described earlier. However, it is also true that many of the author-entered pro and con phrases were not present within the contents of a review: only 34,8% and 23,9% of the normalized forms of the gold standard keyphrases could be matched to some part of the reviews they were assigned to, in the case of mobile phone and the movie reviews, respectively. Owing to this fact, the best performance that could be imagined was an F-score of 51.7 and 38.6 for the two domains. These theoretic F-scores could be achieved by a system that returns all those phrases from the set of gold standard that can be found within the review, but no false positives otherwise. Note that this kind of hypothetic behavior is highly optimistic, as such a system would also need to be able to exactly predict the varying number of opinion phrases to return from review to review (depending on the size of the subset of the gold standard phrases that belongs to the review).

The above-mentioned examples and the performance scores shown in Table 5.4 and 5.6 – compared to those in Table 4.7, which include the results of the best-performing systems on the shared task on extracting keyphrases from scientific articles – suggest that opinion phrase extraction is more difficult compared to scientific keyphrase extraction. This observation being true, we shall see – from our other experiments involving human insight to the evaluation – that the quality of the phrases extracted from product reviews is still of practical use.

Domain adaptation

Most of the standard keyphrase extraction systems use supervised learning techniques. As such they assume the availability of an abundant amount of labeled training data originating from the same target domain as test instances do. Since labeled data for arbitrary target domains are not necessarily accessible, domain adaptation techniques (as described in Section 2.3.2) might be applied in such cases.

Thus we conducted experiments in order to see the applicability of domain adaptation in the opinion phrase extracting task. Classification instances were characterized by the same combined feature set as that used in our previously described experiments. The feature space was then extended based on the feature augmentation technique [29], which employs a simple transformation of the original feature space in order to improve the classification performance of classifiers under domain adaptation settings. The results of the domain adaptation experiments were obtained using cross validation and they are present in Table 5.9 and 5.10, regarding the domains of mobile phones and movies as target domains, respectively.

Rows marked as *Target* refer to the kind of evaluation where training instances originated from the very same target domain as test documents did. This case is thus equivalent to the simple and idealistic case where standard supervised methods are applied in the presence of abundant training data samples from the target domain. Our previous experiments involved models that were trained using this very assumption, so these results are essentially the same as those in Table 5.4 and 5.6

Method	P@5	R@5	F@5	P@10	R@10	F@10	P@15	R@15	F@15
Target	14.8	15.7	15.3	10.4	22.0	14.1	8.0	25.4	12.2
Source	3.5	3.7	3.6	3.6	7.7	4.9	3.6	11.4	5.4
Mixed	11.1	11.8	11.5	8.0	16.9	10.8	11.1	11.8	11.5
Mixed+DA	12.7	13.4	13.0	8.6	18.3	11.7	6.7	21.2	10.2

Table 5.9: Domain adaptation results where the domain of mobile phones is the target domain

Method	P@5	R@5	F@5	P@10	R@10	F@10	P@15	R@15	F@15
Target	10.0	15.8	12.2	7.0	21.9	10.6	5.3	24.6	8.7
Source	3.2	5.0	3.9	2.7	8.5	4.1	2.4	11.2	3.9
Mixed	6.5	10.3	8.0	4.6	14.6	7.1	3.7	17.4	6.1
Mixed+DA	7.2	11.3	8.8	5.0	15.8	7.7	4.0	18.6	6.5

Table 5.10: Domain adaptation results where the domain of movies is the target domain

for the two domains.

We also evaluated the performance of those models which had access exclusively to source domain documents over training, i.e. the Maximum Entropy classifier did not see any evidence from the target domain during the training phase. These (pessimistic) evaluation settings, assuming the total absence of target data for training, are present in the rows denoted by *Source* in Table 5.9 and 5.10. This setting obviously violates the assumption employed by machine learning algorithms, i.e. that training and test data samples originate from the same underlying probability distribution. As this criterion is not met in this case, results are expected to severely decline compared to the performance of models trained in a standard supervised fashion.

The results in rows *Mixed* refer to the evaluation when predictions were performed based on models that were trained on 10% of the target domain and all of the source domain documents. This means that the training instances were heavily biased towards the source domain as there were approximately ten times as many source domain training instances available for tuning the model parameters as target domain instances. During the evaluations denoted by *Mixed+DA*, the training and testing sets were defined in the exact same manner as that for the *Mixed* experiments. The only difference was that the former model used an augmented feature space, inspired by the results presented in [29]. This kind of feature augmentation-based domain adaptation is briefly outlined in Section 5.4.

As expected, the results of intra-domain evaluations (i.e. using the same domains during training and testing) act as an upper bound for the performance of all the other results, when models were trained on less target domain data. It is because of the differences in the underlying probability densities of the source and target domains, which causes training data to become "contaminated" when source domain instances are added to the small number of target training instances (see rows *Mixed* in Table 5.9 and 5.10).

It is also unequivocal from the results of the rows *Source* of Table 5.9 and 5.10 that training

a model just on one source domain (without any target domain instances) and evaluating it on a different target domain will cause a severe drop in performance. Despite the serious decline in the results with the latter settings, giving a small set of target domain documents (having a size of just 10% of the size of the source domain documents) yields much better results. Lastly, employing the feature-augmentation technique results in comparable results to the setting when the training phase had access to all the training instances (see rows marked as *Target* and *Mixed+DA*).

5.4 Related work

There have been many studies on opinion mining [64, 91, 108, 112]. Within the area of opinion mining, our approach relates to previous studies on the extraction of reason for opinions in the form of opinion phrases. Most of the previous papers treated the task of mining reasons from product reviews as one of identifying sentences that express the author's negative or positive feelings, e.g. [48, 94]. Our approach is clearly distinct from previous opinion mining systems, as we proposed a keyphrase extraction techniques-inspired solution to extract opinion phrases, instead of recognizing sentences or clauses that might contain some opinion.

This thesis differs in important aspects even from the frequent pattern mining-based approach of Hu and Liu [49], as they treated the main task of mining opinion features on the basis of a product type (i.e. based on a collection of reviews) and not on the level of individual reviews as we did. Even if an opinion feature phrase is feasible for a given product type, it is not necessary the case that all of its occurrences in any reviews are accompanied with sentiments expressed towards it (e.g. *The phone comes in red and black colors*, where *color* could be an appropriate product feature).

A similar task to pro and con extraction gathers the key aspects from document sets, which has also gained interest recently [20, 64, 104]. Existing aspect extraction systems first identify a number of aspects throughout the whole review set, then they automatically assign items from this pre-recognized set of aspects to each unseen review. Hence, they work at the corpus level and restrict themselves to just using a pre-defined number of aspects. Sullivan [104] treated the task as a multi-label document classification problem and trained binary classifiers for a manually selected set of aspects. Branavan et al. [20] introduced a generative Bayesian topic model to identify several aspects at the corpus level and assigned them to reviews. Liu and Seneff [64] extracted phrases similar to our method, but their goal was to learn sentiment scoring, exploiting adverbs which modify the identified keyphrases. The approach presented in this thesis differs from the other studies in the sense that it searches for the reason phrases themselves, review by review, instead of multi-labeling some aspects.

The work of Kim and Hovy [53] lies probably the closest to ours. They addressed the task of extracting con and pro sentences, i.e. the sentences on why the reviewers liked or disliked a product. They also remarked that such pro and con expressions can differ from positive and negative opinion

expressions as factual sentences can also be reason sentences (e.g. "*Video drains battery.*"). The main difference is thus that they extracted sentences, while our target was the extraction of phrases.

Domain adaptation is the other line of research this chapter is related to. As stated earlier, abundant training examples are not necessarily available from a single domain (product type in the case of opinion phrase extraction), so domain adaptation techniques might be useful in the detection of opinion phrases. Given two sets of instances, $\mathcal{D}_S \in \mathbb{R}^{m_S}$ and $\mathcal{D}_T \in \mathbb{R}^{m_T}$, the former standing for the source domain, the latter being the target domain, with the $|\mathcal{D}_S| \gg |\mathcal{D}_T|$ inequality holding for the distinct domains.

As a possible solution for domain adaptation, the authors of [28] proposed an approach which learns three separate models, one for source-specific, target-specific and general information as well. They also report that the use of EM for the training of the models can be computationally expensive. Although the feature augmentation technique presented in [29] uses an intuition similar to that of [28] (i.e. the existence of source-, target specific and general information), it is much simpler as it learns one model including both source and target domain instances in an extended feature space, instead of learning three models simultaneously. Here, the original feature space is mapped to a higher-dimension space, so that source and target domains and general information are incorporated. In order to achieve this, the mapping Φ_S or Φ_T is employed to every instance $\mathbf{x}$ from the original feature space, depending on whether the original vector $\mathbf{x}$ is representing a source or a target domain instance, respectively. The two mappings have the form $\Phi_S(\mathbf{x}) =< \mathbf{x}, \mathbf{x}, \mathbf{0} >$ and $\Phi_T(\mathbf{x}) =< \mathbf{x}, \mathbf{0}, \mathbf{x} >$, where $\mathbf{0}$ is the null vector.

5.5 Summary of thesis results

In this chapter, the author introduced a novel keyphrase extraction approach for the extraction of pro and con opinion phrases taken from product reviews. Related to his publications [6, 12], the author regards the following results as his main contributions to the research topic:

1. Extension of standard keyphrase extraction to opinion phrase extraction The hypothesis that the underlying principles of extracting opinion phrases from product reviews are analogous to those of extracting keyphrases from non-opinionated utterances – such as scientific publications – was confirmed. The extra-textual information sources, introduced in Chapter 4, for improving the performance of keyphrase extraction also proved beneficial, implying their domain-independence and wide-range applicability. Furthermore, successful ways of adapting the standard keyphrase extraction systems to the specialized task of opinion phrase extraction were also suggested. Our analysis also revealed that the suggested extensions (such as the incorporation of SentiWordNet) are able to improve model performances by a statistically significant amount.

2. Creation of a manually annotated opinion phrase corpus In order to make empirical verification possible, a new dataset comprising of approximately 4,000 product reviews originating from different product domains (i.e. mobile phones and movies) was released for public use. A small (25-element) subset of the mobile phone-domain reviews was assigned multiple reader-assigned opinion phrases, which permits a detailed analysis of opinion phrase extraction.

3. Applying domain adaptation in opinion phrase extraction Opinion phrase extraction was investigated from a domain adaptation point of view. Our experiments provided evidence that satisfactory results can be obtained even in the absence of abundant training data from the target domain. The basic idea of treating opinion phrases in a similar way to scientific keyphrases raises the question of whether domain adaptation methods works in the aspect of scientific articles and product reviews as well. Although these two genres seem to be more different from each other than two sets of reviews on different product families, we find it as one possible way to extend this work to thoroughly examine this particular question.

Chapter 6

Applications of keyphrase extraction

This chapter proposes novel techniques for the task of assigning keyphrases for document subcorpora. The task of determining keyphrases for a subcorpora differs from the generation of keyphrases for single documents, i.e. instead of generating keyphrases on a per document basis, keyphrases need to be determined for multiple documents at a time.

Besides suggesting a solution for the above-mentioned task, we present an empirical validation of the applicability of the single-document keyphrase extraction techniques introduced earlier by their integration into a document collection clustering and visualization framework.

6.1 Motivation

As stated earlier in this thesis, keyphrases can support a range of NLP tasks, including information retrieval and summarization. It can also be argued, however, that when someone searches for some relevant content in a large document set, simply knowing all the keyphrases at the level of individual documents might not be helpful on its own. In such situations, the presence of keyphrases at the sub-corporal level could be useful, making it possible to first perform our search for document subsets. Keyphrases at the level of subcorpora can also provide a better understanding of the topical relations of documents within some corpora.

It will be shown throughout this chapter that having access to the single document keyphrases of some document collection can be utilized to provide keyphrases at sub-corporal levels based on information theoretic grounds. We will also propose a method that can identify reasonable (i.e. thematically coherent) subcorpora within a document collection, based on the assumption that documents with a substantial overlap in their topics will also share important keyphrases.

Determining multi-document keyphrases can also help single-document keyphrase assignment as well, i.e. the keyphrases extracted for a document subset can be treated as likely proper keyphrases for all the documents that comprise the document subset. Assuming this 'topic-aware' property of

the multi-document keyphrases, those topically-related keyphrases that are not otherwise present in some document can be assigned to them.

Furthermore, the clustering and visualization of document sets can be a highly effective and intuitive technique for knowledge discovery [25, 35]. Assigning keyphrases to a cluster of documents makes the understanding of and navigation across these documents much easier for humans. This chapter first proposes an entire framework for identifying thematically coherent document subsets and assigning multi-document keyphrases to them and finally offers a visualization tool for the easier processing of document collections.

6.2 Multi-document keyphrase generation

When determining multi-document keyphrases, we will focus on the more frequent and realistic scenario where keyphrases are not present by default for the documents comprising some corpora. Standard approaches for this task would be based on a bag-of-words model and apply some information theoretical metric to rank the candidate phrases [71]. In our study, instead of employing a bag-of-words approach, we investigated the possible selection of multi-document keyphrase candidates that rely on single-document keyphrase extraction techniques, such as those that were outlined in the previous chapters.

For an empirical evaluation, we decided to assign keyphrases for the workshop papers of ACL Anthology and assess how well they describe the theme of a workshop. As the absolute human evaluation of keyphrases is highly subjective, we asked four researchers who study computational linguistics to compare the quality of the keyphrases of the two approaches with each other. As described at earlier points of the thesis, the automatic evaluation of keyphrases is usually based on strict string matching, which handles semantically (even closely) related phrases as a mismatch if their surface forms differ. We experimented with several automatic evaluation methods for the scientific domain and we shall introduce a procedure for evaluating the multi-document (i.e. workshop level) keyphrases against the original call for papers of the workshops.

6.2.1 Candidate selection and representation

Similar to the case of single document keyphrase extraction, candidate selection plays a key role in multi-document keyphrase extraction. Our general multi-document keyphrase extraction framework first extracts a set of keyphrase candidates, then ranks these candidates on the basis of information theoretic measures. Supposing that we wish to assign keyphrases to l documents at a time (indexed from 1 to l), – using the notation applied in Section 1.1– we defined two strategies for the extraction of multi-document keyphrase candidates, i.e. we treated those phrases as candidates during our experiments which were members of the set

- $\bigcup_{i \in \{1,\ldots,l\}} C_i$, i.e. the union of all the keyphrase candidates belonging to the set of l documents
- $\bigcup_{i \in \{1,\ldots,l\}} K_i$, i.e. the union of all the phrases that were regarded as a single-document keyphrase by our keyphrase extraction model for some of the l documents.

The former approach will be referred as *Baseline*, and the latter as *SDK* (referring to the fact that the candidate phrases were derived from Single-Document Keyphrases). The sets C_i and K_i were defined in a similar way as that described in Chapter 4. The minor differences are discussed below.

Improper keyphrases that were easily recognizable even by their surface forms – like those containing non-English characters and those being shorter than 3 characters – were omitted from the list of single-document candidates. This kind of reduction step favored the baseline system which, unlike the other approach, did not employ any semantical ranking or pre-selection of the keyphrase candidates and often treated rare, but topically unrelated tokens as highly discriminative and thus were worthy of being selected as multi-document keyphrases.

The single-document keyphrase extraction model differed from the one introduced in Chapter 4 in that it had access to more documents for training its parameters. This time the entire 244-document dataset (i.e. both the 144-document train and 100-document test subsets) of the SemEval-2 shared task on scientific keyphrase extraction [55] could be employed for training purposes without the risk of learning a model which overfits, as we were evaluating our approach on documents that were disjoint from that dataset. Additional documents from the 211-document NUS Keyphrase Corpus [88] were also utilized during the training of the parameters of our single-document keyphrase extraction model.

Relying on one of the multi-document keyphrase candidate generation strategies, our system ranked each candidates based on their information gain scores [26]. Information gain for a candidate phrase c, a document set $\mathcal{D}$ and $\mathcal{D}_S \subset \mathcal{D}$ was calculated by the formula

$$IG(\mathcal{D}, \mathcal{D}_S, c) = H(p_{\mathcal{D}}(\mathcal{D}_S)) - \sum_{x \in \{\mathcal{D}_{c^-}, \mathcal{D}_{c^+}\}} \frac{|x|}{|\mathcal{D}|} H(p_x(\mathcal{D}_S)),$$

where H is the entropy function, $\mathcal{D}_{c^-}$ is the subset of those documents from $\mathcal{D}$ which does not contain c, and $\mathcal{D}_{c^+} = \mathcal{D} \setminus \mathcal{D}_{c^-}$. Entropy measures the uncertainty observable in its argument, i.e.

$$H(p_{\mathcal{D}}(\mathcal{D}_S)) = \frac{|\mathcal{D}_S|}{|\mathcal{D}|} \log \left(\frac{|\mathcal{D}|}{|\mathcal{D} \cap \mathcal{D}_S|} \right) + \frac{|\mathcal{D} \setminus \mathcal{D}_S|}{|\mathcal{D}|} \log \left(\frac{|\mathcal{D}|}{|\mathcal{D} \setminus \mathcal{D}_S|} \right).$$

Lastly, candidates with the highest information gain scores having a higher relative frequency within documents in the set $\mathcal{D}_S$ as opposed to $\mathcal{D} \setminus \mathcal{D}_S$ were treated as multi-document keyphrases.

As for the SDK system, we decided to choose the top-15 single-document keyphrases, as false positive predictions are more prevalent above this threshold. This strategy can result in a maximum of $15|\mathcal{D}_S^{(i)}|$ potential keyphrase candidates for subcorpus $\mathcal{D}_S^{(i)}$ of cardinality $|\mathcal{D}_S^{(i)}|$ in the unlikely case that all the top-ranked keyphrases of the individual documents in $\mathcal{D}_S^{(i)}$ were distinct. This theoretical scenario is rather unlikely, especially if the documents in $\mathcal{D}_S^{(i)}$ are topically related to each other, which makes it likely that the individual documents comprising $\mathcal{D}_S^{(i)}$ have at least a few keyphrases in common.

6.3 Experiments and discussion

Now, we present the dataset that was used in our experiments on multi-document keyphrase extraction and also the results achieved with the Baseline and SDK systems.

6.3.1 Dataset

The growing academic interest in the analysis and processing of scientific literature is reflected by the fact that an entire workshop [4] was devoted to it on the 50^{th} anniversary ACL conference. In that workshop, the authors of [98] introduced the corpus on the previous ACL proceedings, which served as a basis for our experiments.

As we wished to find a way to assign keyphrases to thematically coherent subgroups of some document collection, we decided to focus on the part of the ACL Anthology Corpus [98] which contained ACL workshop papers. The reason we just included workshop papers for our experiments was due to the fact that conference workshops tend to be inherently homogeneous in their topic selection; i.e. they tend to focus on some particular, clearly distinguishable area of the larger scientific community, such as *parsing*, *machine translation* or *sentiment analysis* in the case of NLP-related workshops.

However, there were workshops that we felt important to remove as their areas of interest were too broad to handle the papers that were accepted for them as one topically coherent set of documents. The elimination of workshops from the database included proceedings like those of *Empirical Methods of Natural Language Processing* (also known as *EMNLP*), which used to be listed earlier among workshops in the ACL Anthology and which has a topic coverage that is too heterogeneous. The papers suggested for omission were determined by two computational linguistic experts whose inter annotator agreement in terms of accuracy and κ-coefficient was 94.6% and 0.667, respectively, which is to be regarded as a substantial agreement. The κ-coefficient measures the extent of agreement rate between annotators by ruling out the chance agreement between them and is calculated by the formula $\kappa = \frac{P(a)-P(e)}{1-P(e)}$, where $P(a)$ is the agreement rate between the annotators and $P(e)$ is the agreement rate that would be expected by chance.

Total workshop papers	1946
Total distinct workshops	125
Total workshop papers excluded	411 (21.12%)
Total workshops excluded	15 (8.00%)
Average papers per (non-excluded) workshops	13.95 ± 8.10

Table 6.1: Statistics of the workshops present in the ACL Anthology Corpus taken from the 6-year timespan that our experiments focused on

6.3.2 Evaluation

During our experiments, we conducted both human and automated evaluations, the details of which we will present below.

Human evaluation

Creating an exhaustive list with all the keyphrases that can be accepted for a workshop would clearly be a difficult (if not impossible) task. In the absence of a reliable list of this kind, automatic evaluation – whether a suggested keyphrase is a true positive or a false positive – cannot be easily decided. As domain experts can readily decide if a phrase is related to a topic, we decided to conduct evaluation based on human inspection as well.

As a result, 4 researchers working in NLP were hired to make decisions about the sets of keyphrases that were generated by the Baseline and the SDK approaches. The annotators were provided with the sets of keyphrases that were assigned to those workshops of the dataset that were held between the years 2000 and 2005 (inclusive). The manually evaluated subcorpus consisted of 110 workshops with a total of 1,535 documents from the entire corpus, as can be seen in Table 6.1.

Annotators were given the top-3-ranked keyphrases for each workshop taken from both of the system outputs and given the name of the workshop in question (e.g. *ACL-SIGLEX Workshop on Deep Lexical Acquisition*). The task of the annotators was then to make one of the following decisions:

- *Positive draw* or D^+ when both sets of keyphrases might be equally helpful in finding a particular workshop as the keyphrases returned are closely related to the topics of that workshop.
- *Negative draw* or D^- when both sets of keyphrases are of no use; that is, neither of them would be helpful at all if they were looking for the particular workshop.
- *Win* when they were confident that the set of keyphrases they were shown first for a particular workshop was superior in quality to that of the set of keyphrases that was presented to them afterwards.

The order in which the outputs of the two systems were presented to the annotators was shuffled randomly from workshop to workshop. This way, it was impossible for annotators to systematically

	Accuracy	κ-statistic
Annotator_1	80 (72.7%)	0.65
Annotator_2	91 (82.7%)	0.69
Annotator_3	75 (68.2%)	0.48
Annotator_4	74 (67.3%)	0.44

Table 6.2: Annotator agreement rates against the final assessment annotation decisions

	D^+	D^-	Win_{SDK}	Win_{BL}
Annotator_1	8 (7.3%)	12 (10.9%)	63 (57.3%)	27 (24.5%)
Annotator_2	4 (3.6%)	12 (10.9%)	62 (56.4%)	32 (29.1%)
Annotator_3	19 (17.3%)	9 (8.2%)	55 (50.0%)	27 (24.5%)
Annotator_4	17 (15.5%)	16 (14.5%)	54 (49.1%)	23 (20.1%)
Final assessment	1 (0.9%)	10 (9.1%)	**69 (62.7%)**	30 (27.3%)
Automatic evaluation	0 (0.0%)	18 (16.4%)	49 (44.5%)	43 (39.1%)

Table 6.3: The class distribution of the annotation types of the individual annotators and that of the merged final assessments

favor the output of a particular approach during their decision making process. As a consequence of this strategy, there was need for an automatic post-processing step on the annotators' decisions. During this phase, decisions marked as *Win* were automatically divided into two further subcategories, namely Win_{SDK} and Win_{BL}, depending on which system was actually favored by the annotator.

In order to create a final assessment, the individual decisions of the annotators were merged by simply choosing their most frequent decision for each workshop. There were only 9 cases of ties when trying to decide the majority annotation simply by counting, where final decisions were made by revisiting those test cases. This way, a final assessment of decisions was made for each of the 110 human-evaluated workshops based on the independent decisions of 4 human expert annotators. The agreement rates of the four annotators against their combined decisions are listed in Table 6.2, while Table 6.3 shows the distribution of the annotation decisions for each annotator and the combined annotation as well. Due to the commonly accepted interpretation of κ-statistics, the annotators' agreement rates is to be regarded as either moderate or substantial.

As can be seen in Table 6.3, taking the majority vote of the annotators' decisions was useful in the sense that decisions became less ambiguous as D^+ (i.e. tie annotations) almost entirely vanished. In the same table we also notice that – based on the final assessment of annotations – multi-document keyphrases produced by our proposed approach may be regarded as better than those produced by the baseline method for over 62% of the workshops. Keyphrases assigned to sample workshops by both the baseline and the SDK methods are shown in Table 6.4, which – in accordance with the human experts' decisions – also seem to reflect the superiority of the proposed method over the baseline approach.

Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization	
fluency	*machine translation evaluation*
automatic scores	*automatic evaluation*
rouge	*MT evaluation*
Multilingual Question Answering	
correct answer	*QA system*
answer type	*question answering*
monolingual systems	*answering system*
Information Retrieval with Asian Languages	
term frequency	*information retrieval*
retrieval system	*representative keywords*
document frequency	*semantic indexing*
Web as Corpus	
web corpus	*Web as corpus*
wacky project	*search engine*
wacky	*corpus data*

Table 6.4: Sample outputs generated by the two approaches for various workshops of the ACL Anthology Corpus, the baseline keyphrases and the single document-based keyphrases on the left and right hand sides, respectively

Automated evaluation

Besides human evaluations, an automated evaluation was carried out as well, where experiments were conducted on the same workshop data as those for the human evaluations, i.e. the ones that were held between the years 2000 and 2005 (inclusive) and were not judged to be too general in their topic. In order to measure the quality of the workshop-level keyphrases, the original call for papers (CFPs) of the workshops were crawled from the Web, the contents of which served as the basis of comparison for the extracted workshop-level keyphrases.

Using the basic information retrieval techniques described in [71], the quality of each system was measured in the following way. Two vectors were created for the two approaches, both incorporating dimensions for the 1-and 2-grams of those phrases that could be regarded as keyphrase candidates (as described in Section 6.2.1) of the call for papers. For the automatic decision of which systems' output should be regarded as better for a particular workshop, two meta-document vectors were created for the two systems, having non-zero entries just for the top-3 keyphrases. These meta-documents functioned as query vectors and the one which had the greater cosine similarity to the CFP-based prototype vector of the given workshop was selected.

In order to prioritize via term importance within the documents, a tf-weighting of the phrases was used. For the workshop-level meta-vectors, the *tf* term was calculated as the weighted relative frequency of the candidates across all the documents belonging to the workshop. Expressed in formal

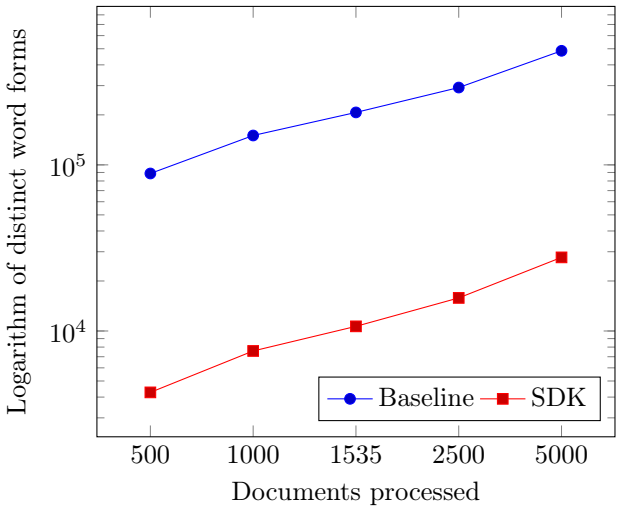


Figure 6.1: The growth in the number of distinct multi-document keyphrase candidate forms as a function of the documents processed

terms, the baseline method was preferred for a workshop i if

$$\frac{x_{CFP,i}^{\intercal} x_{baseline,i}}{\| x_{CFP,i} \| \| x_{baseline,i} \|} > \frac{x_{CFP,i}^{\intercal} x_{SDK,i}}{\| x_{CFP,i} \| \| x_{SDK,i} \|}.$$

This way, we favored a baseline approach if its prototype vector had a bigger cosine similarity to the call-for-paper vector than the prototype vector belonging to the *SDK* approach had. In this kind of evaluation, D^- decisions were equivalent to the situation where neither of the top-3 ranked keyphrases intersected the CFP-based prototype vector, thus resulting in a 0 similarity. In the last row of Table 6.3, we see that this latter kind of evaluation came up more frequently by the automatic method than by humans, but we should add that a keyphrase that is not present in the CFP of a workshop is not necessarily worthless for a given workshop. Equal but non-zero similarities would have yielded D^+ annotations for workshops, but this situation never occurred. In the remaining cases, the Win_{SDK} decision was chosen during the automated evaluation phase.

The method that we proposed – i.e. to rely just on the best-ranked document-level keyphrases and not on all the keyphrase candidates of the individual documents when performing keyphrase extraction for multiple documents – has various advantages. The quality of the keyphrases determined this way is not only superior to the baseline approach based on both human and automatic evaluations, but the size of the distinct word forms (i.e. the vocabulary) – from which keyphrases of document subsets are finally selected – is also reduced by several orders of magnitude (see Figure 6.1). The smaller vocabulary naturally made multi-document keyphrase extraction less resource-intensive and faster without any loss in the quality of the extracted keyphrases.

6.4 Keyphrase-based similarity graph built from documents

In order to represent the inter-document relations among documents, a similarity graph structure was defined that was also used for the fast detection of subcommunities within the corpus. Next, we will describe the structure of the keyphrase similarity graph of documents.

Keyphrases extracted from the individual papers were used as an input for the construction of a similarity graph which served as the basis of the visualization framework. $G_{n,t} = (V, E_n, w_t)$ was defined as a weighted graph of documents, where $E_n = \{(u, v) : v \in neigh(u, n) \vee u \in neigh(v, n)\}$ and $neigh(u, n)$ is a function which returns the set of the n vertices that are closest to vertex u based on the similarity measure w_t.

The similarity measure $w_t(u, v)$ assigns a positive similarity score to documents u and v by comparing the overlap between their top-t keyphrases that best describe them. Values n and t are thus hyperparameters that can be adjusted in our application to see their effects on the connectedness of the document graph.

Since pairs of documents can have multiple keyphrases in common, the calculation of their aggregated similarity scores can be performed in one of several ways (which can be adjusted in the applet). For a similarity graph $G_{n,t}$, the similarity of documents u and v is 0 if the two documents have no keyphrases in common; otherwise it is aggregated due to one of the following strategies, by calculating

1. the Jaccard or Dice similarity between them, using the formulae $\frac{A \cap B}{A \cup B}$ and $\frac{2|A \cap B|}{|A|+|B|}$, respectively,
2. the cosine similarity of the two documents based on their top-t ranked keyphrases,
3. $\sum_{k \in A \cap B} p(k, u)p(k, v)$,
4. $\min_{k \in A \cap B}(p(k, u), p(k, v))$,
5. $\max_{k \in A \cap B}(p(k, u), p(k, v))$,

where sets A and B consist of the top-t ranked keyphrases of documents u and v, respectively and $p(k, u)$ is the probability that is assigned to the event that phrase k is a proper keyphrase of document u.

An analysis of document sets can naturally benefit from knowing which documents are related thematically, so documents sharing similar contents can be grouped together. The assignment of documents to thematically related subcorpora can be sometimes performed automatically or semi-automatically, e.g. scientific papers published at the very same workshop can be assumed to be strongly related to each other. However, documents assigned to differently named workshops on similar topics (such as *Workshop on Computational Approaches to Subjectivity and Sentiment Analysis* and *Workshop on Sentiment Analysis where AI meets Psychology*) should not really be

differentiated from each other. For this reason, we experimented with automated ways of partitioning the document-similarity graph into thematically related subgraphs by employing a modularity maximization strategy, which we will introduce next.

6.4.1 Modularity-driven community detection

The community detection algorithm employed to partition the document-similarity graph relies on the maximization of Newman's modularity [87], which is a measure that qualifies the appropriateness of a graph partitioning, and it can be defined by the following formula:

$$Q = \frac{1}{2m} \sum_{i,j} \Big(A_{i,j} - \frac{k_i k_j}{2m} \Big) \delta(c_i, c_j),$$

where the summation is performed over all *possible* edges (i.e. all pairs of vertices i and j), the value $A_{i,j}$ is simply an element of the adjacency matrix representation of the given graph, m is the total number of edges present, the fraction in the sum is the expected number of edges between nodes i and j, finally, the function δ is the Kronecker delta, which takes the value 1 if its arguments are in the same cluster, and 0 otherwise. Intuitively, what modularity measures for a given partitioning of a graph is the difference between the fraction of intra-community edges and the expected fraction of intra-community edges in the graph with the same number of vertices and edges, but with its edges rewired randomly.

Modularity as an objective function to be maximized is appealing for community detection algorithms, but, it has been shown that its maximization is strongly $\mathcal{NP}$-complete [21]. For this reason, several methods, ranging from simulated annealing to spectral optimization, as well as greedy methods have been developed to approximate that partitioning of a graph, which gains the highest modularity value. For a comprehensive survey on the topic, see the work of Fortunato [40].

Even though spectral optimization tend to yield better results [40], its application is often unfeasible for larger graphs. As our intention was to be able to deal with possibly massive data collections, it was essential to keep the computation requirements relatively low. Blondel et al. [18] introduced a method which greedily approximates that partitioning of a graph that has the highest modularity. The proposed iterative method works in a bottom-up manner, starting from the state in which all the vertices of the graph form a separate community. In the following steps, vertices are moved into a community in such a way that their replacement should yield the best local increase in the modularity.

Moving a vertex i to a community C has a two-fold effect: first, node i will produce an increase in modularity due to its edges that pass within its new community C, but it will also cause modularity to decrease, as its edges that pass through its previous community by then will no longer contribute to modularity (because of the Kronecker-delta part in the definition of modularity). The increase in

modularity when one vertex i is moved to community C can be expressed as:

$$\Delta Q = \left[\frac{\sum_{in} + k_{i,in}}{2m} - \left(\frac{\sum_{tot} + k_i}{2m} \right)^2 \right] - \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m} \right)^2 - \left(\frac{k_i}{2m} \right)^2 \right],$$

where $\sum_{in}$ and $\sum_{tot}$ are the sums of the weights of the links inside C and incident to C, respectively, k_i and $k_{i,in}$ are the sums of the weights edges involving i and having and endpoint in C, respectively, and m is the total edge weighted sum of the graph to be partitioned. When decisions have been made for all the vertices, the communities are collapsed, and treated again as simple vertices of a smaller graph (with its vertex number equaling that of the previously determined number of communities).

This kind of iteration continues until no more gain in modularity can be obtained. The great advantage of this procedure is its time-efficiency on general sparse graphs. Furthermore, the proposed approach can yield a hierarchic community structure of the partitioned graph as the number of iterations performed can influence the coarseness of the partitioning, which can be utilized in the automatic detection of topics in a document set.

6.4.2 Visualization of the document graph

Finally, we implemented a visualization framework which integrates all the previously introduced elements (i.e. the multi-document keyphrase generation submodule and the community detection algorithm) into an application. In order to graphically display the document-similarity graph, we used the force-directed layout technique that is part of TouchGraph [99], a publicly available Java library for visualizing graphs. We extended the functionality of the framework to provide more user interaction, making users able to

1. filter documents for keyphrases
2. filter documents for some meta-data (such as the publication date or the authors of a paper)
3. assign keyphrases for a manually selected subset of a corpus
4. toggle the display of individual documents or thematically-related subcorpora.

Our application supports two means for the determination of the thematically-related subcorpora within some corpus. If the user has some reliable a priori knowledge on how documents in a corpus form subcorpora, this information can be provided for the framework. In the absence of such knowledge, document communities are automatically detected based on the principles that were introduced in Section 6.4.

Subsequently, nodes also get colored to indicate which document belongs to which subcorpora. The framework assigns each subcorpora a color, and nodes get colored based on the assigned color of the subcorpora they belong to. We wanted a coloring that is capable of reflecting the topical

similarity among document subgroups, i.e. we wanted the colors assigned to thematically related subcorpora to be similar. We also wanted the coloring to be efficiently computable as an initialization step for larger document collections as well. To obtain such a coloring, prototype vectors based on the single-document keyphrase-based representation of the documents belonging to subcorpora were created. These prototype vectors were simply taken as the average of those vectors which constituted a subcorpus. Prototype vectors were projected into a 3-dimensional orthogonal subspace capturing as much of the variation of the data as possible by relying on principal component analysis (see Section 2.2.2). The RGB components of the colors assigned to the subcorpora were then determined by comparing the coordinates of the projected prototype vectors and the unit standard basis vectors (i.e. the unit vectors co-directional with the x, y, and z axes of a three dimensional Cartesian coordinate system).

The problem with applying PCA directly to the space of prototype vectors is that it requires the computation of possibly large covariance matrices and their eigenvalue-eigenvector pairs. Even for moderately large vocabulary sizes, these calculations can be expensive from a computational point of view, prohibiting their calculation in real-time applications as a preprocessing step. In order to decrease the computational needs of this step, a random projection was first performed on the data matrix consisting of community prototype vectors. There exist both practical [14] and theoretical [27] results which claim that, under certain conditions, pairwise distances between high-dimensional data points are not much distorted by random projections into a lower-dimensional subspace. Performing such a projection thus has the advantage of approximately preserving the distances between data points and making the computational needs of a PCA performed subsequently less resource-intensive.

Based on the approach introduced in Section 6.2, the framework also determines multi-document keyphrases for the document subcorpora as these sets of multi-document keyprhases can provide additional insights for the users. Finally, to illustrate the importance of nodes within the document graph, the application determines the PageRank [90] values for each vertex. Since the similarity graph $G_{n,t}$ was designed to be sparse – influenced by the parameter of maximal neighbors n – these calculations can be efficiently calculated relying on the use of fast sparse matrix multiplications during the initialization phase of the application. To overcome the so-called "dead end" and "spider trap" effects on the calculation of PageRank values of the nodes, a random walk with teleportation probability $\beta = 0.15$ was employed. In the end, the size of the nodes is influenced by their rank within the graph. A demo of the entire visualization framework can be accessed from the URL `https://www.inf.u-szeged.hu/~berendg/keyphraseViz/`.

6.5 Related work

Although the assignment of representative phrases for (sub)sets of documents can be undoubtedly beneficial, it has been less studied so far compared to the assignment of keyphrases to individual

documents. The common approach for this task is to rank each n-gram in the document set via some information theoretic or statistical (e.g. χ^2) score. The patent [105] also applies a similar approach by using partial mutual information for the determination of keyphrases. Our solution also has an information theoretical basis, but our chief contribution here is that our system exploits deeper positional, linguistic and semantic information concerning the occurrences of the candidate phrases via single-document keyphrase extraction techniques.

Perhaps the most closely related work to ours is the CorePhrase algorithm [45], which was designed to extract keyphrases for document collections relying on a graph structure called Document Index Graph. Although the authors of [45] also focused on multi-document keyphrase extraction, they assumed that "keyphrases exist in the text and are not automatically generated", and that the algorithm "extracts keyphrases from already clustered documents". In this thesis, we focused on the more frequent and realistic case where there are no manually assigned keyphrases available for the documents, nor the partitioning of the corpus into thematically related subgroups is known in advance.

Keyphrase extraction when performed on (scientific) documents may also be beneficial for research on (scientific) trend detection, as applied in [44], where the changes in *focus*, *technique* and *domain*-related expressions of scientific publications in the field of computational linguistics were analyzed over time. Our study follows this line of research as keyphrases describing a cluster (document set) can provide clues for trend detection.

The graph-based approach to generate extractive summaries from scientific articles in [95] utilized the so-called *Citation Summary Network* to create extractive summaries of scientific articles. They employed their approach not only for single documents but for scientific topics, i.e. multiple documents from the same area as well. Their study differed from ours as they treated the topics to be summarized and the assignment of documents to those topics as an input for their algorithm, while our framework did not have access to these information. Also, summaries can be beneficial in becoming familiar with a topic from a glance, but they can be hardly utilized to reveal the intra-topic document relations or the relatedness of different topics to each other. Furthermore, as they used citation analysis, it makes the approach dependent on the availability of citation information, which makes their application primarily suitable for handling scientific documents. Our approach, however, only relies on keyphrases, which can be interpreted and automatically determined for genres different from that of scientific publications as well.

Topic models such as Latent Dirichlet Allocation [17] provide an efficient way to analyze document sets. In their model, documents are treated as a mixture of topics where each topic has a distribution over the vocabulary of words. Although topic models are able to reveal general trends and identify topics based on word usage of documents, they do not really reveal how documents are organized within each topic and it is also unclear how different topics are related to each other. The generative framework employed in LDA is also very sensitive to the setting of its hyperparameters [116].

Eisenstein et al. [35] introduced *TopicViz*, a Latent Dirichlet Allocation (LDA)-based document visualization system, which can be interpreted as a visually-aided information retrieval system. There are two basic differences between their approach and ours. First, they relied on topic models, whereas we employed graph partitioning in order to automatically determine document subtopics. Second, in their study they manually identified the topics determined by LDA [17], whereas we let the automatically detected communities "speak for themselves", i.e. the most informative sets of keyphrases of size 3 were determined based on information theoretic grounds. Our proposed method did not need to know the number of topics to be identified in advance and its time requirements are also more favorable compared to the training of topic models, i.e. it can be performed on the fly during the initialization of our application. A further possible advantage of our approach compared to other LDA-style models is that topic models tend to be trained at the level of single tokens, whereas our approach can easily extract informative noun phrases and multi-word expressions as it operates at the level of n-grams.

In scientific document set visualization, citation analysis is often taken into account. The explicit relations among documents like citations can be naturally included in our graph-based approach as well (which is not straightforward in LDA-based solutions). However, citation-based methods have the limitation that they are mostly useful for scientific document sets where citation information is accessible.

6.6 Summary of thesis results

In this chapter, the author showed how single document keyphrases can be incorporated in various solutions that are available to alleviate difficulties users encounter when they need to find relevant contents on some topic. This way, the author gave a complete framework in his dissertation starting with the generation of keyphrases for documents of various genres, which can be then utilized in the visualization of corpora. Related to his publications [9, 10], the author regards the following results as his main contributions to the field:

1. Proposing a method for assigning keyphrases to document subcorpora Empirical evaluation suggested that the task of multi-document keyphrase extraction could benefit from the knowledge conveyed by the single-document keyphrases. Keyphrases assigned to thematically-related document subcorpora by the proposed approach were found to be as useful as the baseline approach in more than 70% of the test cases.

2. Applying single-document keyphrases-based document representation We showed that such a representation of documents which relies on single-document keyphrases can be as expressive as traditional n-gram based representations. Furthermore, storing documents in the proposed way can reduce the amount of memory needed for storing document collections by several

orders of magnitude compared to that for standard approaches. The reason is that the overall vocabulary for a corpus which is based on the keyphrases of the documents which comprise it is much smaller than the dictionary which contains all the distinct n-grams of the corpus. The radical reduction in the size of the dictionary also speeds up the calculation of document pair similarities, hence it can be integrated into real-time applications.

3. Introduction of the keyphrase similarity graph-based clustering and visualization Using the proposed single-document keyphrases-based document representation and the fast community detection algorithm (based on the work of Blondel et al. [18]) together allowed us to integrate all the modules into an efficient corpus visualization framework. We believe that such frameworks can provide a useful means for knowledge discovery.

Chapter 7

Summary

7.1 Summary in English

The main goal of this thesis was to demonstrate techniques for the generation of keyphrases extracted from textual documents of various types, i.e. news articles, scientific documents and product reviews. The proposed solutions take into account the specificities of the domains and widely rely on extra-textual world knowledge by utilizing Wikipedia. Here, we briefly summarize the main results of Chapters 3-6.

7.1.1 Keyphrase Generation from Newswire

In Chapter 3, we demonstrated our system constructed for the assignment of keyphrases for the news articles comprising the archive of the news portal Origo. This task had special characteristics, as we not only had to handle the morphological richness of the Hungarian language, but also provide that the keyphrases assigned to the news articles behave coherently at the level of the entire document collection (e.g. by providing that they do not contain synonymous expressions). A further specificity of the task was that we had to take special care of the so-called *abstract keyphrases*, i.e. keyphrases that are not otherwise present in the document for which they need to be assigned. Combining our approaches into a framework, we managed to achieve a document-level precision of 75.44%, which was well beyond the prior expectations of the employees of Origo. Related to his publication [38], the author regards the following as his main contributions to the research topic:

- Introduction of a ranking procedure for selecting the most likely keyphrases
- Assigning abstract keyphrases to documents based on definitions derived from Wikipedia
- Various ways of assigning abstract keyphrases to documents based on the link structure of Wikipedia

7.1.2 Keyphrase Extraction from Scientific Documents

In Chapter 4, we proposed novel ways of exploiting extra-textual information during the extraction of keyphrases. Besides these features being useful in the domain of scientific publications – as Chapter 5 verified it – they tend to be successfully applicable for different domains as well, implying their wide-range applicability. The proposed method performs competitively with state-of-the-art systems according to our evaluations performed on multiple datasets. One of the main advantages of the proposed method is that even though it relies on Wikipedia similar to some other approaches, it does not require a full index to be created from all the textual contents of Wikipedia, as it relies only on its category structure. Nevertheless our approach requires less resources, it can still perform competitively with other methods. Related to his publications [7, 8, 84], the author regards the following as his main contributions to the research topic:

- Extension of existing keyphrase candidate set filtering techniques
- Introduction of a condensed representations for sequential features
- Utilization of extra-textual information for representing keyphrase candidates

7.1.3 Opinion Phrase Extraction

In Chapter 5, we verified our hypothesis that keyphrase extraction techniques can be employed to the task of determining important aspects of product reviews, i.e. *pro and con* expressions. Besides pointing out the applicability of standard keyphrase extraction approaches for that task, we suggested a handful of domain-specific features. Incorporating these features into standard keyphrase extraction frameworks resulted in significant gains in performance. In our experiments, we also investigated the subjective nature of judging the appropriateness of keyphrases. Domain differences among product reviews were demonstrated via cross-product experiments within the domain of product reviews. It was also shown how the severe performance drop in the quality of the extracted keyphrases in such cases can be lessened by using adaptation methods. Related to his publications [6, 12], the author regards the following as his main contributions to the research topic:

- Extension of standard keyphrase extraction to opinion phrase extraction
- Creation of a manually annotated opinion phrase corpus
- Verification of the applicability of domain adaptation techniques in opinion phrase extraction

7.1.4 Applications of keyphrase extraction

In Chapter 6, we demonstrated the application possibilities of keyphrase identification approaches. The chapter proposed a solution for the assignment of keyphrases for subcorpora on the basis of information theoretic considerations and the sets of keyphrases determined for the individual documents.

Furthermore, in that chapter it was also shown how topics can be inferred from document-level keyphrases and then be visualized based on the overlap between the documents in a text collection. Related to his publications [9, 10], the author regards the following results as his main contributions to the field:

- Proposing a method for assigning keyphrases to document subcorpora
- Applying single-document keyphrases-based document representation
- Introducing the keyphrase similarity graph-based clustering and visualization framework

7.2 Summary in Hungarian

A disszertáció elsődleges célja a különböző doménekből (újsághírek, tudományos publikációk, termékvéleményezések) származó dokumentumok kulcsszavainak automatikus meghatározására alkalmas algoritmusok bemutatása. Az egyes domének szövegeinek feldolgozására javasolt eljárások figyelembe veszik azok sajátosságait, valamint elmondható róluk, hogy nagyban támaszkodnak a kulcsszavazandó dokumentumon kívülről származó külső információkra. A következőkben rövid bemutatásra kerülnek a tézispontok főbb eredményei.

7.2.1 Újsághírek kulcsszavazása

A 3. fejezetben az Origo hírportál szöveges archívumában található dokumentumokhoz történő kulcsszavak hozzárendelésére irányuló munkáját mutatja be a szerző. Az itt bemutatott kulcsszavazási feladat több sajátossággal is rendelkezett. A kulcsszavazó eljárásnak egyrészt tudnia kellett kezelni a magyar nyelv morfológiai gazdagságából adódó sajátosságokat. Ezen felül külön fontossággal bírt, hogy a kinyert kulcsszavak ne csupán az egyes dokumentumok leírására legyenek alkalmasak, hanem a teljes hírarchívum tekintetében is koherensen viselkedjenek (pl. a szinonim jelentésű kulcsszavak elkerülésével). További jellemzője volt a feladatnak azon ún. *absztrakt kulcsszavak* kezelésének a kiemelt fontossága. Az absztrakt kulcsszavak úgy képesek egy-egy dokumentum tartalmának összefoglalására, hogy abban nem találhatók meg. A fejezetben bemutatott rendszer kiértékelése alapján a meghatározott kulcsszavak a dokumentumok 75.44%-ában érték el a kívánatos minőséget, mely eredmény jelentősen meghaladta az Origo hírportál által támasztott előzetes elvárásokat. Korábbi publikációja [38] alapján a szerző a tézis legfontosabb saját eredményeinek a következőket tekinti:

- A kulcsszójelöltek rangsorolására létrehozott módszer
- Absztrakt kulcsszavak meghatározása a Wikipédiából kinyert definíciók alapján
- Absztrakt kulcsszavak meghatározása a Wikipédia linkstruktúrája alapján

7.2.2 Tudományos publikációk kulcsszavazása

A 4. fejezetben újszerű szövegen kívüli jellemzőkre támaszkodó kulcsszavazó modelljét mutatta be a szerző. A bevezetett jellemzők nem csupán a tudományos publikációk doménjén voltak alkalmazhatók, hanem – ahogy azt az 5. fejezetbeli alkalmazásuk mutatta – doménfüggetlen tulajdonsággal bírtak, ami széleskörű alkalmazhatóságukat vetíti előre. A javasolt modell hasonlóan vagy jobban teljesített 2 standard benchmark tudományos publikációkat tartalmazó adatbázison is a jelenleg elérhető kulcsszavazó rendszerekhez képest.

A szerző kulcsszavazási megoldásában dokumentumon kívüli információra is támaszkodott, melynek fő forrása a Wikipédia volt. Korábbi munkákban ugyan találkozhattunk már hasonló elképzelésekkel, fontos különbség azonban, hogy míg a szerző kizárólag a Wikipédia kategóriastruktúráját fölhasználva épített be külső tudást rendszerébe, addig más munkák a Wikipédia összes szöveges tartalmának feldolgozását és indexelését tették szükségessé, jóval erőforrásigényesebbé téve ezáltal a hasonló megközelítéseket. Korábbi publikációi [7, 8, 84] alapján a szerző a tézis legfontosabb saját eredményeinek a következőket tekinti:

- A korábbi kulcsszójelölt-állítási stratégiák kiterjesztése
- Szekvenciákat kódoló jellemzők alternatív reprezentációjának bevezetése
- Szövegen kívüli információk kiaknázása

7.2.3 Véleménykifejezések kinyerése

Az 5. fejezetben igazolást nyert a szerző azon hipotézise, mely szerint a kulcsszókinyerési technikák adaptálhatók a véleménykifejezések kinyerésére irányuló feladat megoldása során. Mindezek mellett a szerző a kinyert véleménykifejezések minőségét jelentősen javítani képes doménspecifikus jellemzőket is bemutatott tézisében. A különböző terméktípusok véleménykifejezéseinek kinyerésének átjárhatóságát is vizsgálta a szerző, mely során doménadaptációs kísérleteket hajtott végre. Korábbi publikációi [6, 12] alapján a szerző a tézis legfontosabb saját eredményeinek a következőket tekinti:

- Standard kulcsszavazási feladat kiterjesztése véleménykifejezések kinyerésére
- Termékvéleményezésekből és a hozzájuk tartozó véleménykifejezésekből álló korpusz létrehozása
- Doménadaptációs vizsgálatok a különböző termékcsoportok véleménykifejezéseinek kinyerése közötti átjárhatóság biztosítására

7.2.4 Kulcsszókinyerés alkalmazásai

A 6. fejezetben kulcsszókinyerő rendszerének végalkalmazásait ismertette a szerző. A fejezetben bemutatásra került egy információelméleti alapokon nyugvó módszer dokumentumcsoportok kulcsszavainak meghatározására. A fejezet rámutatott arra is, hogy miként lehet korpuszon belüli témákat detektálni a dokumentumok kulcsszavai közötti átfedés vizsgálata útján, illetve hogy miként lehet ezt az információt korpuszvizualizáció során fölhasználni. Korábbi publikációi [9, 10] alapján a szerző a tézis legfontosabb saját eredményeinek a következőket tekinti:

- Dokumentumhalmazok kulcsszavainak meghatározására irányuló eljárás
- Dokumentumok kulcsszóalapú reprezentációjának bevezetése
- Kulcsszóhasonlósági gráf alapján történő klaszterezés és korpuszvizualizáció

Bibliography

[1] Anna Babarczy, Bálint Gábor, Gábor Hamp, and András Rung. Hunpars: a rule-based sentence parser for Hungarian. In *Proceedings of the 6th International Symposium on Computational Intelligence*, 2005.

[2] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May 2010. European Language Resources Association (ELRA). ISBN 2-9517408-6-7. URL `http://www.lrec-conf.org/proceedings/lrec2010/pdf/769_Paper.pdf`.

[3] Alexandra Balahur, Ester Boldrini, Andres Montoyo, and Patricio Martinez-Barco, editors. *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA 2.011)*. Association for Computational Linguistics, Portland, Oregon, June 2011. URL `http://www.aclweb.org/anthology/W11-17`.

[4] Rafael E. Banchs, editor. *Proceedings of the ACL-2012 Special Workshop on Rediscovering 50 Years of Discoveries*. Association for Computational Linguistics, Jeju Island, Korea, July 2012. URL `http://www.aclweb.org/anthology/W12-32`.

[5] Ken Barker and Nadia Cornacchia. Using noun phrase heads to extract document keyphrases. In *Proceedings of the 13th Biennial Conference of the Canadian Society on Computational Studies of Intelligence: Advances in Artificial Intelligence*, AI '00, pages 40–52, London, UK, UK, 2000. Springer-Verlag. ISBN 3-540-67557-4. URL `http://dl.acm.org/citation.cfm?id=647461.726264`.

[6] Gábor Berend. Opinion expression mining by exploiting keyphrase extraction. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 1162–1170, Chiang Mai, Thailand, November 2011. Asian Federation of Natural Language Processing. URL `http://www.aclweb.org/anthology/I11-1130`.

[7] Gábor Berend. Exploiting extra-textual information in keyphrase extraction. *Natural Language Engineering*, page to appear, 2014.

[8] Gábor Berend and Richárd Farkas. SZTERGAK: Feature engineering for keyphrase extraction. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 186–189, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/S10-1040`.

[9] Gábor Berend and Richárd Farkas. Keyphrase-driven document visualization tool. In *The Companion Volume of the Proceedings of IJCNLP 2013: System Demonstrations*, pages 17–20, Nagoya, Japan, October 2013. Asian Federation of Natural Language Processing. URL `http://www.aclweb.org/anthology/I13-2005`.

[10] Gábor Berend and Richárd Farkas. Single-document keyphrase extraction for multi-document keyphrase extraction. *Computación y Sistemas*, 17(2):179–186, 2013.

[11] Gábor Berend and Veronika Vincze. How to evaluate opinionated keyphrase extraction? In *Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis*, pages 99–103. Association for Computational Linguistics, 2012.

[12] Gábor Berend, István T. Nagy, György Móra, and Veronika Vincze. Inter-domain opinion phrase extraction based on feature augmentation. In *Proceedings of the Second Student Research Workshop associated with RANLP 2011*, pages 41–47, Hissar, Bulgaria, September 2011. RANLP 2011 Organising Committee. URL `http://aclweb.org/anthology/R11-2006`.

[13] Adam L. Berger, Vincent J. Della Pietra, and Stephen A. Della Pietra. A maximum entropy approach to natural language processing. *Comput. Linguist.*, 22(1):39–71, March 1996. ISSN 0891-2017. URL `http://dl.acm.org/citation.cfm?id=234285.234289`.

[14] Ella Bingham and Heikki Mannila. Random projection in dimensionality reduction: Applications to image and text data. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '01, pages 245–250, New York, NY, USA, 2001. ACM. ISBN 1-58113-391-X. doi: 10.1145/502512.502546. URL `http://doi.acm.org/10.1145/502512.502546`.

[15] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. ISBN 0387310738.

[16] Bo-Christer Björk, Annikki Roos, and Mari Lauri. Scientific journal publishing: yearly volume and open access availability. *Inf. Res.*, 14(1), 2009. URL `http://dblp.uni-trier.de/db/journals/ires/ires14.html#BjorkRL09`.

[17] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003. ISSN 1532-4435. URL `http://dl.acm.org/citation.cfm?id=944919.944937`.

[18] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008+, July 2008. ISSN 1742-5468. doi: 10.1088/1742-5468/2008/10/p10008. URL `http://dx.doi.org/10.1088/1742-5468/2008/10/p10008`.

[19] Adrien Bougouin, Florian Boudin, and Béatrice Daille. TopicRank: Graph-based topic ranking for keyphrase extraction. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 543–551, Nagoya, Japan, October 2013. Asian Federation of Natural Language Processing. URL `http://www.aclweb.org/anthology/I13-1062`.

[20] S.R.K. Branavan, Harr Chen, Jacob Eisenstein, and Regina Barzilay. Learning document-level semantic properties from free-text annotations. In *Proceedings of ACL-08: HLT*, pages 263–271, Columbus, Ohio, June 2008. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P/P08/P08-1031`.

[21] Ulrik Brandes, Daniel Delling, Marco Gaertler, Robert Görke, Martin Hoefer, Zoran Nikoloski, and Dorothea Wagner. On modularity – np-completeness and beyond, 2006.

[22] Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. Class-based n-gram models of natural language. *Comput. Linguist.*, 18(4):467–479, December 1992. ISSN 0891-2017. URL `http://dl.acm.org/citation.cfm?id=176313.176316`.

[23] Chris Buckley and Ellen M. Voorhees. Retrieval evaluation with incomplete information. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, pages 25–32, New York, NY, USA, 2004. ACM. ISBN 1-58113-881-4. doi: 10.1145/1008992.1009000. URL `http://doi.acm.org/10.1145/1008992.1009000`.

[24] Alexander Budanitsky and Graeme Hirst. Evaluating WordNet-based measures of lexical semantic relatedness. *Comput. Linguist.*, 32(1):13–47, mar 2006. ISSN 0891-2017. doi: 10.1162/coli.2006.32.1.13. URL `http://dx.doi.org/10.1162/coli.2006.32.1.13`.

[25] Jason Chuang, Christopher D. Manning, and Jeffrey Heer. "Without the clutter of unimportant words": Descriptive keyphrases for text visualization. *ACM Trans. on Computer-Human Interaction*, 19:1–29, 2012. URL `http://vis.stanford.edu/papers/keyphrases`.

[26] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory 2nd Edition (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, July 2006. ISBN 0471241954.

[27] Sanjoy Dasgupta and Anupam Gupta. An elementary proof of a theorem of johnson and lindenstrauss. *Random Structures & Algorithms*, 22(1):60–65, 2003. ISSN 1098-2418. doi: 10.1002/rsa.10073. URL `http://dx.doi.org/10.1002/rsa.10073`.

[28] Hal Daumé, III and Daniel Marcu. Domain adaptation for statistical classifiers. *J. Artif. Int. Res.*, 26:101–126, May 2006. ISSN 1076-9757. URL http://portal.acm.org/citation.cfm?id=1622559.1622562.

[29] Hal Daume III. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P07-1033.

[30] Jeffrey Dean and Sanjay Ghemawat. MapReduce: Simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, January 2008. ISSN 0001-0782. doi: 10.1145/1327452.1327492. URL http://doi.acm.org/10.1145/1327452.1327492.

[31] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, 39(1):1–38, 1977.

[32] Zhuoye Ding, Qi Zhang, and Xuanjing Huang. Keyphrase extraction from online news using binary integer programming. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 165–173, Chiang Mai, Thailand, November 2011. Asian Federation of Natural Language Processing. URL http://www.aclweb.org/anthology/I11-1019.

[33] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2000. ISBN 0471056693.

[34] Ted Dunning. Accurate methods for the statistics of surprise and coincidence. *Comput. Linguist.*, 19(1):61–74, March 1993. ISSN 0891-2017. URL http://dl.acm.org/citation.cfm?id=972450.972454.

[35] Jacob Eisenstein, Duen Horng Chau, Aniket Kittur, and Eric Xing. Topicviz: interactive topic exploration in document collections. In *CHI '12 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '12, pages 2177–2182, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1016-1. doi: 10.1145/2212776.2223772. URL http://doi.acm.org/10.1145/2212776.2223772.

[36] Folke Eisterlehner, Andreas Hotho, and Robert Jäschke, editors. *ECML PKDD Discovery Challenge 2009 (DC09)*, volume 497 of *CEUR-WS.org*, September 2009.

[37] Richárd Farkas, Veronika Vincze, István Nagy, Róbert Ormándi, György Szarvas, and Attila Almási. Web based lemmatisation of named entities. In *Proceedings of the 11th International Conference on Text, Speech and Dialogue*, pages 53–60, 2008.

[38] Richárd Farkas, Gábor Berend, István Hegedűs, András Kárpáti, and Balázs Krich. Automatic free-text-tagging of online news archives. In *Proceedings of the 2010 conference on ECAI 2010: 19th European Conference on Artificial Intelligence*, pages 529–534, Amsterdam, The Netherlands, The Netherlands, 2010. IOS Press. ISBN 978-1-60750-605-8. URL `http://dl.acm.org/citation.cfm?id=1860967.1861071`.

[39] C. Fellbaum. *WordNet: An Electronic Lexical Database*. Language, Speech and Communication. Mit Press, 1998. ISBN 9780262061971. URL `http://books.google.at/books?id=Rehu8OOzMIMC`.

[40] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75–174, 2010.

[41] Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In *In Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 1606–1611, 2007.

[42] John Gantz and David Reinsel. The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east. *IDC iView: IDC Analyze the Future*, 2012.

[43] Maria Grineva, Maxim Grinev, and Dmitry Lizorkin. Extracting key terms from noisy and multi-theme documents. In *18th International World Wide Web Conference (WWW2009)*, April 2009. URL `http://data.semanticweb.org/conference/www/2009/paper/67`.

[44] Sonal Gupta and Christopher Manning. Analyzing the dynamics of research by extracting key aspects of scientific papers. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 1–9, Chiang Mai, Thailand, November 2011. Asian Federation of Natural Language Processing. URL `http://www.aclweb.org/anthology/I11-1001`.

[45] Khaled M. Hammouda, Diego N. Matute, and Mohamed S. Kamel. Corephrase: keyphrase extraction for document clustering. In *Proceedings of MLDM*, pages 265–274, 2005.

[46] Kazi Saidul Hasan and Vincent Ng. Conundrums in unsupervised keyphrase extraction: Making sense of the state-of-the-art. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pages 365–373, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=1944566.1944608`.

[47] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[48] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD

'04, pages 168–177, New York, NY, USA, 2004. ACM. ISBN 1-58113-888-1. doi: http://doi.acm.org/10.1145/1014052.1014073. URL http://doi.acm.org/10.1145/1014052.1014073.

[49] Minqing Hu and Bing Liu. Mining opinion features in customer reviews. In *Proceedings of the 19th national conference on Artifical intelligence*, AAAI'04, pages 755–760. AAAI Press, 2004. ISBN 0-262-51183-5. URL http://portal.acm.org/citation.cfm?id=1597148.1597269.

[50] Anette Hulth. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, EMNLP '03, pages 216–223, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1119355.1119383. URL http://dx.doi.org/10.3115/1119355.1119383.

[51] Arif E. Jinha. Article 50 million: an estimate of the number of scholarly articles in existence. *Learned Publishing*, 23(3):258–263, July 2010. ISSN 0953-1513. doi: 10.1087/20100308. URL http://dx.doi.org/10.1087/20100308.

[52] Daniel Jurafsky and James H. Martin. *Speech and Language Processing (2nd Edition) (Prentice Hall Series in Artificial Intelligence)*. Prentice Hall, 2nd edition, 2008. ISBN 0131873210.

[53] Soo-Min Kim and Eduard Hovy. Automatic identification of pro and con reasons in online reviews. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 483–490, Sydney, Australia, July 2006. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P/P06/P06-2063.

[54] Su Nam Kim and Min-Yen Kan. Re-examining automatic keyphrase extraction approaches in scientific articles. In *Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications*, MWE '09, pages 9–16, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-60-2. URL http://dl.acm.org/citation.cfm?id=1698239.1698242.

[55] Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. Semeval-2010 task 5: Automatic keyphrase extraction from scientific articles. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, SemEval '10, pages 21–26, Morristown, NJ, USA, 2010. ACL. URL http://portal.acm.org/citation.cfm?id=1859664.1859668.

[56] Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. Automatic keyphrase extraction from scientific articles. *Language resources and evaluation*, 47(3):723–742, 2013.

[57] Dan Klein and Christopher D. Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st ACL*, pages 423–430, 2003. doi: 10.3115/1075096.1075150. URL http://www.aclweb.org/anthology/P03-1054.

[58] Roman Klinger, Katrin Tomanek, and Roman Klinger. Classical probabilistic models and conditional random fields, 2007.

[59] András Kuba, András Hócza, and János Csirik. Pos tagging of Hungarian with combined statistical and rule-based methods. In *Proceedings of the 7th International Conference on Text, Speech and Dialogue*, pages 113–120, 2004.

[60] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, CA, 2001. URL `citeseer.ist.psu.edu/lafferty01conditional.html`.

[61] Thomas K Landauer and Susan T. Dutnais. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, pages 211–240, 1997.

[62] David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. RCV1: A new benchmark collection for text categorization research. *J. Machine Learning Research*, 5:361–397, 2004.

[63] Bing Liu. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data (Data-Centric Systems and Applications)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. ISBN 3540378812.

[64] Jingjing Liu and Stephanie Seneff. Review sentiment scoring via a parse-and-paraphrase paradigm. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 161–169, Singapore, August 2009. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/D/D09/D09-1017`.

[65] Zhiyuan Liu, Peng Li, Yabin Zheng, and Maosong Sun. Clustering to find exemplar terms for keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 257–266, Singapore, August 2009. URL `http://www.aclweb.org/anthology/D/D09/D09-1027`.

[66] Zhiyuan Liu, Wenyi Huang, Yabin Zheng, and Maosong Sun. Automatic keyphrase extraction via topic decomposition. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 366–376, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=1870658.1870694`.

[67] Patrice Lopez and Laurent Romary. HUMB: Automatic key term extraction from scientific articles in GROBID. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, SemEval '10, pages 248–251, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=1859664.1859719`.

[68] Patrice Lopez, Laurent Romary, et al. GRISP: A massive multilingual terminological database for scientific and technical domains. In *LREC 2010*, 2010.

[69] Abdulhussain E. Mahdi and Arash Joorabchi. A citation-based approach to automatic topical indexing of scientific literature. *J. Inf. Sci.*, 36(6):798–811, December 2010. ISSN 0165-5515. doi: 10.1177/0165551510388080. URL `http://dx.doi.org/10.1177/0165551510388080`.

[70] Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA, 1999. ISBN 0-262-13360-1.

[71] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schtze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008. ISBN 0521865719, 9780521865715.

[72] Cameron Marlow, Mor Naaman, Danah Boyd, and Marc Davis. Position paper, tagging, taxonomy, flickr, article, toread. In *Collaborative Web Tagging Workshop at WWW2006*, May 2006. URL `http://www.danah.org/papers/WWW2006.pdf`.

[73] Luís Marujo, Márcio Viveiros, and João Paulo Neto. Keyphrase cloud generation of broadcast news. In *INTERSPEECH*, pages 2393–2396, 2011.

[74] Luís Marujo, Anatole Gershman, Jaime G. Carbonell, Robert E. Frederking, and João Paulo Neto. Supervised topical key phrase extraction of news stories using crowdsourcing, light filtering and co-reference normalization. In *LREC*, pages 399–403, 2012.

[75] Luís Marujo, Wang Ling, Anatole Gershman, Jaime G. Carbonell, João Paulo Neto, and David Martins de Matos. Recognition of named-event passages in news articles. In *COLING (Demos)*, pages 329–336, 2012.

[76] Luís Marujo, Ricardo Ribeiro, David Martins de Matos, João Paulo Neto, Anatole Gershman, and Jaime G. Carbonell. Key phrase extraction of lightly filtered broadcast news. In *TSD*, pages 290–297, 2012.

[77] Andrew Kachites McCallum. MALLET: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

[78] Olena Medelyan and Ian H. Witten. Thesaurus based automatic keyphrase indexing. In *Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries*, JCDL '06, pages 296–297, New York, NY, USA, 2006. ACM. ISBN 1-59593-354-9. doi: 10.1145/1141753.1141819. URL `http://doi.acm.org/10.1145/1141753.1141819`.

[79] Olena Medelyan, Eibe Frank, and Ian H. Witten. Human-competitive tagging using automatic keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods*

in Natural Language Processing, pages 1318–1327, Singapore, August 2009. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/D/D09/D09-1137`.

[80] Rada Mihalcea and Paul Tarau. Textrank: Bringing order into texts. In *Proceedings of EMNLP*, volume 4, page 275. Barcelona, Spain, 2004.

[81] Gilad Mishne. AutoTag: a collaborative approach to automated tag assignment for weblog posts. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 953–954, New York, NY, USA, 2006. ACM Press. ISBN 1595933239. doi: 10.1145/1135777.1135961.

[82] Thomas M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1st edition, 1997. ISBN 0070428077, 9780070428072.

[83] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective (Adaptive Computation and Machine Learning series)*. The MIT Press, August 2012. ISBN 0262018020. URL `http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0262018020`.

[84] István Nagy T., Gábor Berend, and Veronika Vincze. Noun compound and named entity recognition and their usability in keyphrase extraction. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*, pages 162–169, Hissar, Bulgaria, September 2011. RANLP 2011 Organising Committee. URL `http://aclweb.org/anthology/R11-1023`.

[85] Roberto Navigli and Simone Paolo Ponzetto. Babelrelate! a joint multilingual approach to computing semantic relatedness. In *AAAI Conference on Artificial Intelligence*, 2012. URL `http://www.aaai.org/ocs/index.php/AAAI/AAAI12/paper/view/5112`.

[86] João Paulo Neto, Hugo Meinedo, and Márcio Viveiros. A media monitoring solution. In *ICASSP*, pages 1813–1816. IEEE, 2011. ISBN 978-1-4577-0539-7. URL `http://dblp.uni-trier.de/db/conf/icassp/icassp2011.html#NetoMV11`.

[87] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2):026113+, February 2004. doi: 10.1103/PhysRevE.69.026113. URL `http://dx.doi.org/10.1103/PhysRevE.69.026113`.

[88] Thuy Dung Nguyen and Min-Yen Kan. Keyphrase extraction in scientific publications. In *Proceedings of the 10th international conference on Asian digital libraries: looking back 10 years and forging new frontiers*, ICADL'07, pages 317–326, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 3-540-77093-3, 978-3-540-77093-0. URL `http://dl.acm.org/citation.cfm?id=1780653.1780707`.

[89] Thuy Dung Nguyen and Minh-Thang Luong. WINGNUS: Keyphrase extraction utilizing document logical structure. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, SemEval '10, pages 166–169, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=1859664.1859699`.

[90] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: Bringing order to the web., November 1999. URL `http://ilpubs.stanford.edu:8090/422/`. Previous number = SIDL-WP-1999-0120.

[91] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *EMNLP '02: Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, pages 79–86, Morristown, NJ, USA, 2002. Association for Computational Linguistics. doi: http://dx.doi.org/10.3115/1118693.1118704.

[92] Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. WordNet::Similarity: Measuring the relatedness of concepts. In *Demonstration Papers at HLT-NAACL 2004*, HLT-NAACL–Demonstrations '04, pages 38–41, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=1614025.1614037`.

[93] Simone Paolo Ponzetto and Michael Strube. Deriving a large scale taxonomy from Wikipedia. In *AAAI*, volume 7, pages 1440–1445, 2007.

[94] Ana-Maria Popescu and Oren Etzioni. Extracting product features and opinions from reviews. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 339–346, Vancouver, British Columbia, Canada, October 2005. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/H/H05/H05-1043`.

[95] Vahed Qazvinian, Dragomir R. Radev, Saif M. Mohammad, Bonnie Dorr, David Zajic, Michael Whidby, and Taesun Moon. Generating extractive summaries of scientific paradigms. *J. Artif. Int. Res.*, 46(1):165–201, January 2013. ISSN 1076-9757. URL `http://dl.acm.org/citation.cfm?id=2512538.2512543`.

[96] Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th international joint conference on Artificial intelligence - Volume 1*, IJCAI'95, pages 448–453, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc. ISBN 1-55860-363-8, 978-1-558-60363-9. URL `http://dl.acm.org/citation.cfm?id=1625855.1625914`.

[97] Ivan A. Sag, Timothy Baldwin, Francis Bond, Ann A. Copestake, and Dan Flickinger. Multiword expressions: A pain in the neck for NLP. In *Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing

'02, pages 1–15, London, UK, UK, 2002. Springer-Verlag. ISBN 3-540-43219-1. URL http://dl.acm.org/citation.cfm?id=647344.724004.

[98] Ulrich Schäfer, Jonathon, and Stephan Oepen. Towards an ACL anthology corpus with logical document structure. an overview of the ACL 2012 contributed task. In *Proceedings of the ACL-2012 Special Workshop on Rediscovering 50 Years of Discoveries*, pages 88–97, Jeju Island, Korea, July 2012. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W12-3210.

[99] Alexander Shapiro. Touchgraph, 2001. URL http://sourceforge.net/projects/touchgraph/.

[100] Konstantin Shvachko, Hairong Kuang, Sanjay Radia, and Robert Chansler. The Hadoop Distributed File System. In *Proceedings of the 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*, MSST '10, pages 1–10, Washington, DC, USA, 2010. IEEE Computer Society. ISBN 978-1-4244-7152-2. doi: 10.1109/MSST.2010.5496972. URL http://dx.doi.org/10.1109/MSST.2010.5496972.

[101] Sanjay Sood, Sara Owsley, Kristian Hammond, and Larry Birnbaum. TagAssist: Automatic tag suggestion for blog posts. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM 2007)*, 2007. URL http://icwsm.org/papers/2--Sood-Owsley-Hammond-Birnbaum.pdf.

[102] Michael Strube and Simone Paolo Ponzetto. WikiRelate! computing semantic relatedness using Wikipedia. In *AAAI'06: Proc. of the 21st National Conference on Artificial Intelligence*, pages 1419–1424, 2006.

[103] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: A core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 697–706, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-654-7. doi: 10.1145/1242572.1242667. URL http://doi.acm.org/10.1145/1242572.1242667.

[104] Todd Sullivan. Pro, con, and affinity tagging of product reviews. Technical Report 224n, Stanford CS, 2008.

[105] Arungunram C. Surendran. Multi-document keyphrase extraction using partial mutual information. Patent, 05 2010. URL http://www.patentlens.net/patentlens/patent/US_7711737/en/. US 7711737.

[106] György Szarvas, Richárd Farkas, and András Kocsor. A multilingual named entity recognition system using boosting and c4.5 decision tree learning algorithms. *DS2006, LNAI*, 4265:267–278, 2006.

[107] M. Tatu, M. Srikanth, and T. D'Silva. RSDC'08: Tag recommendations using bookmark content. In *Proceedings of the ECML PKDD Discovery Challenge 2008*, 2008.

[108] Ivan Titov and Ryan McDonald. A joint model of text and aspect ratings for sentiment summarization. In *Proceedings of ACL-08: HLT*, pages 308–316, Columbus, Ohio, June 2008. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P/P08/P08-1036`.

[109] Takashi Tomokiyo and Matthew Hurst. A language model approach to keyphrase extraction. In *Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment - Volume 18*, MWE '03, pages 33–40, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1119282.1119287. URL `http://dx.doi.org/10.3115/1119282.1119287`.

[110] Kristina Toutanova and Christopher D. Manning. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora*, EMNLP '00, pages 63–70, Stroudsburg, PA, USA, 2000. ACL. doi: http://dx.doi.org/10.3115/1117794.1117802. URL `http://dx.doi.org/10.3115/1117794.1117802`.

[111] Peter Turney. Learning algorithms for keyphrase extraction. *Information Retrieval*, 2:303–336, 2000.

[112] Peter Turney. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 417–424, 2002.

[113] Peter Turney. Coherent keyphrase extraction via web mining. In *Proceedings of IJCAI '03*, pages 434–439, 2003.

[114] Rudolf Ungváry and Tamás Radnai. Discover thesauri: State of the art in Hungary. In *Proceedings of 3rd Slovakian-Hungarian Joint Symposium on Applied Machine Intelligence*, 2005.

[115] Ellen M. Voorhees. The TREC-8 question answering track report. In *In Proceedings of TREC-8*, pages 77–82, 1999.

[116] Hanna M. Wallach, David M. Mimno, and Andrew McCallum. Rethinking LDA: Why priors matter. In *NIPS*, pages 1973–1981, 2009.

[117] Xiaojun Wan and Jianguo Xiao. Single document keyphrase extraction using neighborhood knowledge. In *Proceedings of the 23rd national conference on Artificial intelligence - Volume*

2, AAAI'08, pages 855–860. AAAI Press, 2008. ISBN 978-1-57735-368-3. URL `http://dl.acm.org/citation.cfm?id=1620163.1620205`.

[118] David X. Wang, Xiaoying Gao, and Peter Andreae. DIKEA: Domain-independent keyphrase extraction algorithm. In *Proceedings of the 25th Australasian Joint Conference on Advances in Artificial Intelligence*, AI'12, pages 719–730, Berlin, Heidelberg, 2012. Springer-Verlag. ISBN 978-3-642-35100-6. doi: 10.1007/978-3-642-35101-3_61. URL `http://dx.doi.org/10.1007/978-3-642-35101-3_61`.

[119] Ian H. Witten, Gordon W. Paynter, Eibe Frank, Carl Gutwin, and Craig G. Nevill-Manning. Kea: Practical automatic keyphrase extraction. In *ACM DL*, pages 254–255, 1999.

[120] Fei Wu and Daniel S. Weld. Open information extraction using Wikipedia. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 118–127, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=1858681.1858694`.

[121] Zhaohui Wu and C Lee Giles. Measuring term informativeness in context. In *Proceedings of NAACL-HLT*, pages 259–269, 2013.

[122] Shiren Ye, Tat-Seng Chua, and Jie Lu. Summarizing definition from Wikipedia. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 199–207. Association for Computational Linguistics, 2009.

[123] Eric Yeh, Daniel Ramage, Christopher D. Manning, Eneko Agirre, and Aitor Soroa. WikiWalk: Random walks on Wikipedia for semantic relatedness. In *Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing*, TextGraphs-4, pages 41–49, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-54-1. URL `http://dl.acm.org/citation.cfm?id=1708124.1708133`.

[124] Wei You, Dominique Fontaine, and Jean-Paul A. Barthès. An automatic keyphrase extraction system for scientific documents. *Knowl. Inf. Syst.*, 34(3):691–724, 2013.

[125] Torsten Zesch and Iryna Gurevych. Approximate matching for evaluating keyphrase extraction. In *Proceedings of the 7th International Conference on Recent Advances in Natural Language Processing*, pages 484–489, September 2009.