Part of Speech n-Grams for Information Retrieval

UNIVERSITY
of
GLASGOW

Christina Amalia Lioma

Department of Computing Science

University of Glasgow

A thesis submitted for the degree of

Doctor of Philosophy

December 2007

To my family

Acknowledgements

I wish to thank the whole of the Information Retrieval group of the University of Glasgow, and in particular: Iadh Ounis, for giving me the opportunity and means to do research, supervising me, and teaching me valuable lessons; Vassilis Plachouras (now at Yahoo!), Ben He, and Craig Macdonald, for their kind support, guidance and fruitful co-operation; Robert Villa, Sachi Arafat, Hideo Joho, Jana Urban (now at Google), and Frank Hopfgartner, for useful suggestions and their very kind encouragement and support.

Particularly special thanks must go to: Leif Azzopardi, for offering outstanding advice and guidance, refreshing brainstorming, very constructive feedback and the kindest support; Roi Blanco (A Coruña University), for offering very insightful feedback and advice, absolutely invaluable help and support, and kind encouragement through his healing guitar and Monty Pythonesque semantics; Keith van Rijsbergen, for supervising me and guiding me wisely through the good times and the bad, with insightful advice and intellectual stimuli, for his outstanding and always refreshing feedback, for supporting and encouraging me, and most of all for inspiring me with his attitude to research and life. Leif, Roi, Keith, I am deeply grateful to you.

I also wish to thank the committee of my PhD defence, Jian-Yun Nie, Richard Cooper, and Matthew Chalmers, for taking the time to read this thesis and for making very insightful suggestions.

Last but not least, I wish to thank my family, a point of reference, source of inspiration, solace, motivation and unconditional support.

Abstract

The increasing availability of information on the World Wide Web (Web), and the need to access relevant specs of this information provide an important impetus for the development of automatic intelligent Information Retrieval (IR) technology. IR systems convert human authored language into representations that can be processed by computers, with the aim to provide humans with access to knowledge. Specifically, IR applications locate and quantify informative content in data, and make statistical decisions on the topical similarity, or relevance, between different items of data. The wide popularity of IR applications in the last decades has driven intensive research and development into theoretical models of information and relevance, and their implementation into usable applications, such as commercial search engines.

The majority of IR systems today typically rely on statistical manipulations of individual lexical frequencies (i.e., single word counts) to estimate the relevance of a document to a user request, on the assumption that such lexical statistics can be sufficiently representative of informative content. Such estimations implicitly assume that words occur independently of each other, and as such ignore the compositional semantics of language. This assumption however is not entirely true, and can cause several problems, such as ambiguity in understanding textual information, misinterpreting or falsifying the original informative intent, and limiting the semantic scope of text. These problems can hinder the accurate estimation of relevance between texts, and hence harm the performance of an IR application.

This thesis investigates the use of non-lexical statistics by IR models, with the goal to enhance the estimation of relevance between a document and a user request. These non-lexical statistics consist of *part of speech* information. The parts of speech are the grammatical classes of words (e.g., noun, verb). Part of speech statistics are modelled in the form of part of speech (POS) n-grams, which are contiguous sequences of parts of speech, extracted from text.

The distribution of POS n-grams in language is statistically analysed. It is shown that there exists a relationship between the frequency and informative content of POS n-grams. Based on this, different applications of POS n-grams to IR technology are described and evaluated with state of the art systems. Experimental results show that POS n-grams can assist the retrieval process.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 Introduction

This thesis investigates the use of part of speech (POS) n-grams to information retrieval (IR). The main argument of the thesis is that POS n-grams encode grammatical and structural information about language in a shallow way, which can be statistically manipulated to indicate the non-topical informative content of words in a reliable way. Non-topical content refers to how informative a word is in general, and not with respect to a topic.

Two main issues are addressed in this thesis. Firstly, a set theoretical framework is introduced for representing grammatical categories (parts of speech), which are modelled as contiguous sequences (n-grams). Within this framework, it is shown that there exists an approximately directly proportional relationship between the frequency and informative content of POS n-grams, unlike words, for which frequency and informative content are approximately inversely proportional. Based on this finding, it is shown how to derive a non-topical information score for words using exclusively POS n-grams. Secondly, applications of POS n-grams to IR are presented and evaluated. In total, four applications are presented: two of them are direct applications of POS n-gram frequency to IR, namely for query reformulation and index pruning; the other two are applications of the proposed term information score to IR, namely as an alternative to the *inverse document frequency* (IDF) term weight, and also as additional evidence that can enhance overall retrieval performance.

The remainder of this chapter is organised as follows. Section 1.2 presents the motivation of this work. Section 1.3 states the contributions of this work. Section 1.4 gives an overview of the structure of this thesis.

1.2 Motivation

In textual IR, information is usually *identified* and *quantified* using lexical statistics, e.g., word counts. Identifying information allows for decisions to be made about the topical similarity, or *relevance*, between two pieces of text. Quantifying information extends these decisions to how related one piece of text is to another.

The lexical statistics used in IR stem from two observations about language:

- A word occurring very often in a document is likely to indicate the document content.

- A word occurring very often in a general collection of documents is not likely to indicate the content of any document in particular (Sparck Jones, 1972).

On the basis of these observations, different types of lexical statistics are combined to estimate the content of a document automatically. Such lexical statistics are usually the frequency of a word in a document, the number of documents containing a word, and so on. These lexical statistics are the main ingredients of mathematical functions that compute scores for words (term weights). These term weights represent the contribution of a word to the content of a document. The document content is then derived from the term weights of the words occurring in it. The more accurate these term weights, the more accurate the estimation of the document content, and hence the more accurate the estimation of relevance between a document and a query.

The estimation of relevance between a document and a query based on lexical statistics uses mainly *topical information*. For example, given a word A and a word B, the task is to decide how related the topic of A is to the topic of B. This thesis proposes that this estimation can be improved by *non-topical information*, and specifically grammatical and structural statistics of language. For example, given a word A and a word B, the task is to decide if A is generally more informative than B on the basis of their respective grammatical type and the contexts in which they are likely to occur, regardless of their topic. The hypothesis underlying this thesis is that adding a non-topical information layer (grammatical and structural statistics) to the estimation of topical information (lexical statistics) by IR systems can improve retrieval performance.

This thesis models grammatical and structural statistics as POS n-grams, which are contiguous sequences of parts of speech, e.g., determiner-adjective-noun, adjective-noun-verb, noun-verb-adverb, and so on. The motivation for using part of speech information, as opposed to using other types of linguistic information, such as semantic or discourse evidence, is that:

- Parts of speech represent shallow grammatical information, which can indicate to an extent the presence or absence of content. For example, a word is likely to be informative if it is a noun, and similarly not very informative if it is a pronoun. When viewed like this, the indication of presence or absence of content is independent of the exact sense (semantics) of the word. Hence parts of speech capture non-topical information about words, without having semantic or discourse knowledge about their use or context. In this light, POS n-grams can become ‘POS contexts’ for which there is some prior knowledge of content, e.g., POS n-grams containing nouns and verbs are likely to be more informative than POS n-grams containing prepositions and adverbs.

- Parts of speech are a small and finite set of categories, which can be used to annotate text of any domain quickly and relatively accurately (state of the art POS tagging performance approaches $> 90\%$ accuracy). In this respect, parts of speech are better suited for being used in IR than other bigger and open-ended ontologies, such as semantic graphs for example, which can be domain-bounded, and also subject to scalability issues and accuracy $< 90\%$.

Based on the empirical observation that the more frequent these ‘POS contexts’ are, the more informative they actually are, this thesis develops several applications of POS n-grams to IR.

1.3 Thesis statement

The statement of this thesis is that shallow grammatical and structural information about language can be encoded in POS n-grams and used to estimate the non-topical informative content of words in a reliable way. Shallow grammatical information is captured by parts of speech. Structural information is captured by n-gram modelling. Basic principles of linguistic theory are used to rank the informative content of parts of speech. Basic principles of probability theory are used to approximate the informative content in POS n-grams and individual words.

The main contributions of this thesis are the following. A framework is introduced for representing grammatical and structural information about language in a shallow way using POS n-grams. Within this framework, it is shown that there exists an approximately directly proportional relationship between the frequency and informative content of POS n-grams, and also that a term weight can be derived exclusively from POS n-grams. The statistical properties of POS n-grams and of the proposed term weight that is computed from them are examined in a series of thorough experiments on different corpora and settings. In addition, different applications of POS n-grams and of the proposed term weight to IR are presented and evaluated. Experimental evidence shows that the proposed applications can enhance retrieval performance, in different datasets and settings.

1.4 Thesis outline

This thesis is organised as follows.

- Chapter 2, page 8, provides a brief overview of the main concepts of IR. This chapter presents the main processes involved in a standard IR system, the main retrieval models, and issues of IR system operation and evaluation.

- Chapter 3, page 38, provides a brief overview of the main concepts of parts of speech as shallow grammatical categories. A linguistic theory for ranking parts of speech is introduced. This chapter also presents the set theoretical notation of parts of speech proposed in this thesis, and related studies using parts of speech.

- Chapter 4, page 50, provides a brief overview of the main concepts of n-grams, including the notation used in this thesis. This chapter also introduces POS n-grams. Related studies using n-grams and POS n-grams in particular are also presented.

- Chapter 5, page 59, discusses the relationship between the frequency and informative content of POS n-grams, and also introduces the proposed framework for deriving a term information score using POS n-grams. Two alternative term scores are proposed, and different facets of this derivation (assumptions and implications) are discussed.

- Chapter 6, page 76, studies the distribution of POS n-grams and their corresponding POS n-gram based term information score in different corpora. The choice of experimental settings is analysed thoroughly to show that it is not biased. The POS n-gram based term information score is statistically analysed with reference to a lexically based term score (IDF), and the two are shown to be positively correlated.

- Chapter 7, page 99, proposes applications of POS n-grams to IR. Four applications are integrated into the retrieval process and evaluated as part of an IR system: two direct applications of POS n-gram frequency to IR, and two applications of the proposed POS n-gram based term information score to IR. Overall, experimental evidence shows that the proposed applications can enhance retrieval performance.

- Chapter 8, page 143, summarises the contributions and conclusions of this thesis. Limitations of this work are discussed and future research directions are suggested.

1.5 Contributions

The main contributions of this thesis are the following.

- A linguistic theory for ranking parts of speech, namely Jespersen's Rank Theory (Jespersen, 1913, 1929), is used in a principled way in IR. To our knowledge, this is the first time that this theory is used in IR or any other automatic language processing technology.

- Heuristical evidence is presented which suggests that there exists an approximately directly proportional relationship between POS n-gram frequency and informative content. This novel finding is the opposite of what is observed with words, for which the relationship between frequency and informative content is approximately inversely proportional.

- A framework is introduced for deriving an original term information score exclusively from POS n-grams, based on the relationship between POS n-gram frequency and informative content and also on Jespersen's Rank Theory.

- POS n-grams are used, not as a feature for classification, neither to make predictions about the occurrence of parts of speech/words, as has been done so far, but as a feature of non-topical informative content. This is a novel use of POS n-grams.

- The statistical properties of POS n-grams and of the proposed term information score that is computed from them are examined in a series of thorough and unbiased experiments, which include five standard and established collections of different size (totalling >32GB) and domain, three established state of the art POS taggers, and a variation of the n-gram order n between n= 1 - 100. Experimental evidence shows that POS n-grams are distributed similarly in different collections, and that the POS n-gram based term information score is positively correlated to inverse document frequency.

- Four novel applications of POS n-grams to IR are presented and evaluated on standard and established datasets, under default and competitive settings. Experimental evidence shows that retrieval performance can benefit considerably.

1.6 Publications

Parts of this thesis are included in the following publications:

- Lioma & Ounis (2005) *Deploying part-of-speech patterns to enhance statistical phrase-based machine translation resources.* Association for Computational Linguistics (ACL) Workshop on Building and Using Parallel Texts.

- Lioma & Ounis (2006) *Examining the content load of part-of-speech blocks for information retrieval.* Joint Conference of the International Committee on Computational Linguistics and the Association for Computational Linguistics, (COLING/ACL).

- Lioma *et al.* (2006) *University of Glasgow at TREC 2006: experiments in terabyte and enterprise tracks with Terrier.* National Institute of Standards and Technology (NIST) Text REtrieval Conference (TREC).

- Ounis *et al.* (2007) *Research directions in Terrier: a search engine for advanced retrieval on the Web.* Novatica/UPGRADE Special Issue on Web Information Access.

- Lioma & Ounis (2007a) *Extending weighting models with a term quality measure.* Symposium on String Processing and Information Retrieval (SPIRE).

- Lioma & Ounis (2007b) *Light syntactically-based index pruning for information retrieval.* European Conference on Information Retrieval (ECIR).

- Lioma & Ounis (2008) *A syntactically-based query reformulation technique for information retrieval.* Information Processing & Management (IPM).

- Lioma & van Rijsbergen (2008) *Part of speech n-grams and information retrieval.* French Journal of Applied Linguistics (RFLA) Special Issue on Linguistics and Automatic Access to Information.

Chapter 2

Basic concepts of information retrieval

2.1 Introduction

This chapter presents some basic concepts of information retrieval (IR), and the main processes involved in an IR system. Issues affecting the operation and evaluation of IR systems are also introduced. The material in this chapter has been drawn from Baeza-Yates & Ribeiro-Neto (1999); Belew (2000); Lancaster & Fayen (1973); Salton (1971); Salton & McGill (1983); Sparck Jones & Willett (1997); van Rijsbergen (1979), unless otherwise stated.

Section 2.2 presents a general overview of IR, and gives the structure of the rest of this chapter.

2.2 Information retrieval overview

IR investigates the efficient and effective storage and access of information in text, sound, video, images, or other types of data, which can be found stand-alone, in databases, or hypertext networks like the World Wide Web (Web). The increasing widespread of technology for generating and disseminating data has led to an explosion of information availability, rendering the retrieval of relevant information a necessary and cumbersome task. Automatic IR processes address this task by locating and quantifying information in data, and estimating its topical similarity, or *relevance*, to user needs.

A common IR scenario is the following: while performing a task, a user needs to locate information in a repository of documents. In IR, *document* typically refers to any type of data stored in the system (e.g., documents, emails, book chapters). The repository of documents from which information is retrieved is typically referred to as document *collection* (Sparck Jones & van Rijsbergen, 1976). The user's expression of information need is typically referred to as *query*, and usually contains keywords. The user is interested in documents that are relevant to the query. The goal of an IR system is to return all the relevant documents, and no non-relevant documents. The retrieved documents should preferably be ranked with respect to their relevance to the query. The process exemplified in this scenario is referred to as *ad-hoc information retrieval*.

IR systems usually operate in stages. Given a system and a collection of documents, these stages are:

1. The collection from which information is to be retrieved is entered into the IR system. User queries are formulated and entered into the system[1].
2. Queries and documents are transformed into representations that the system can process.
3. Document representations are matched to query representations.
4. Documents matched to queries are returned to the user.

This process can be iterative, i.e., a query can be reformulated and then parts of this process will be repeated. (This process is discussed in Section 2.5.1.)

How users formulate and enter queries to the system, and how retrieved documents are displayed to users are stages that involve human interaction. There is extended research on how these processes can be tailored to user needs and satisfaction (see Marchionini (1995) and Shneiderman (1997) for an introduction). The intermediate stages of the retrieval process, namely how to represent and match documents and queries, are usually fully automatic, involving very little or no human interaction. These automatic processes are the focus of this thesis, in the context of ad-hoc information retrieval from text.

The rest of this chapter is organised as follows: Section 2.3 introduces how IR systems represent documents and queries. Section 2.4 gives an overview of retrieval models for matching documents to queries. Section 2.5 presents techniques often used to enhance retrieval performance. Section 2.6 presents issues

[1] The collection of documents and user queries do not have to be entered into the system simultaneously.

relating to the efficiency of IR systems (e.g., faster processing, or saving memory). Section 2.7 introduces how IR systems are typically evaluated. Section 2.8 summarises and concludes this chapter.

2.3 Query and document representation

Given a query and a collection of documents as input, the IR system creates in an efficient way a representation of this input. Queries and documents are represented in the same way, so that they can be matched later (see Section 2.4 for matching documents to queries). In an experimental situation, queries can be processed at the same time as documents. In an operational situation, usually documents are processed in advance (offline), and queries are processed when they are submitted to the system (online). The document representation techniques described next also apply to query representation.

Extracting a representation of documents consists in splitting the input into tokens, and realising a set of operations to transform the raw tokens into features that can be stored by the system. This process is often referred to as *pre-processing* or *parsing*. The main pre-processing operations are *stopword removal* (or *stopping*) and *stemming*. The output of these operations is stored by the IR system in an *index*. Confusingly, in some IR literature pre-processing is referred to as index term extraction or simply *indexing*, while in other IR literature *indexing* refers to the task of constructing an index after pre-processing (Zobel & Moffat, 2006). In this thesis, indexing refers to the task of constructing an index after pre-processing.

Section 2.3.1 presents an overview of pre-processing, and Section 2.3.2 presents an overview of indexing.

2.3.1 Pre-processing

The IR system input is pre-processed before it is indexed. Typically, pre-processing entails a set of operations, aiming to address the following questions:

- Should very frequent words of very little meaning (e.g., `the`) be indexed?

This point is addressed by stopword removal, which is presented in Section 2.3.1.1.

- Should terms be indexed in their full form (e.g., `medicine, medical`), or should morphological variants be reduced to some base form (e.g., `medic`)?

This point is addressed by stemming, which is presented in Section 2.3.1.2.

- Should terms be lowercased? Should hyphenated terms be considered as one word or two? When HTML documents are being indexed, should the markup tags be indexed? Should terms within such tags be indexed?

The last three questions are of lesser impact to retrieval performance (Zobel & Moffat, 2006), and are usually addressed arbitrarily (Kobayashi & Takeda, 2000).

2.3.1.1 Stopword removal

The aim of stopword removal is to remove commonly occurring words from text. Stopwords are content-poor words, which tend to occur very frequently in many documents, e.g., `the, and, of.` As such, the contribution of stopwords to the content of a document is usually negligible. IR systems tend to remove stopwords altogether for two main reasons:

- Stopwords are non-discriminate words, which do not contribute significantly towards better matching documents to queries.
- Stopwords add considerably to the storage required by the system because they occur in almost all documents.

As a result, removing stopwords can benefit system effectiveness (there is less chance of matching query words to stopwords erroneously), and system efficiency (storing less words requires less resources (e.g., disk space, processing time)). There are exceptions to this, for instance with some queries containing phrases, even stopwords can make an important contribution when matching documents to queries (Zobel & Moffat, 2006).

Stopwords can be manually predefined, or selected according to their frequency on the basis that they occur very frequently in the whole collection. If documents are grammatically annotated, stopwords can also be defined according to grammatical information (e.g., `the` is a determiner, `and` is a coordinating conjunction, `of` is a preposition) (Strzalkowski & Lin, 1997). One of the first widely used IR systems, the SMART system developed at Cornell University, initially used a stopword list of 571 words (Zobel & Moffat, 2006). Today, stopword lists can contain from a few dozens up to approximately over a thousand terms, for English.

The output of stopword removal is a set of words for each document. This is passed onto stemming.

2.3.1.2 Stemming

In an IR system, stemming is used to reduce variant word forms to common 'stems' or base forms, so that morphological variants (e.g., `medical, medicine`) are stored under one entry (e.g., `medic`). This can improve the ability of the system to match query and document vocabulary, by increasing the number of relevant documents retrieved (recall), because it can expand the original query with related word forms (Krovetz, 1993; Porter, 1980; **?**), and also by fetching more relevant documents (precision), because it can promote the more relevant documents to higher ranks than the other retrieved documents (Xu & Croft, 1996). (The notions of recall and precision are explained in Section 2.7.)

In language, the variety in word forms comes from both inflectional and derivational morphology, and stemmers are usually designed to handle both, although in some systems stemming consists solely of handling plurals (Xu & Croft, 1996). Stemmers typically use some surface linguistic information (e.g., removing derivational endings, such as "-ion") and pattern matching rules. Stemming can also be realised by removing word endings or suffixes, using tables of common endings and heuristics about when it is appropriate to remove them.

One of the best known stemmers used in experimental IR systems is the Porter stemmer (Porter, 1980), which iteratively removes endings from a word until termination conditions are met. The Porter stemmer has been criticised for having a number of problems that are also found in other stemmers, in varying degrees (Xu & Croft, 1996):

- The stemming algorithm is not always easy to understand and modify.
- The stemmer makes errors by sometimes being too aggressive in conflation (e.g., `policy` - `police`, `executive` - `execute` are conflated), and by missing others (e.g., `European` - `Europe`, `matrices` - `matrix` are not conflated).
- The stemmer produces stems that are not words and which are not always easy for a user to intepret (e.g., `iteration` produces `iter`, and `general` produces `gener`).

Another stemmer is KSTEM (Krovetz, 1993), which stems words based on machine-readable dictionaries and well-defined rules for inflectional and derivational morphology. Even though KSTEM addresses many of the problems with the Porter stemmer, it does not produce consistently better retrieval performance. KSTEM has been criticised for being heavily dependent on the entries of the dictionary being used, and for being conservative in conflation.

Overall, evaluations of stemming for IR have produced mixed results (Harman, 1987, 1991a,b; Hull, 1996): recall/precision evaluations of the Porter stemmer have shown that it performs at least as well as other stemmers, and at most slightly better than other stemmers (Hull, 1996). Krovetz (1993) and Xu & Croft (1996) have showed small improvement in retrieval performance when using stemming. Today, there is still no clear consensus on the usability of stemming for IR.

The output of stemming is a set of stemmed words, or *terms*[1], for each document. The output of stemming is passed on to indexing.

2.3.2 Indexing

In a standard IR system, after parsing the documents and queries, an index is constructed, so that documents can be retrieved with respect to a query, on the basis of the terms contained in the documents and queries. Broadly speaking, this process consists in storing in the IR system two types of information:

- Straight-forward mappings between terms and documents (e.g., which term occurs in which document). Such mappings typically consist of frequency counts.

- A weight for each term, which represents how much the term contributes to the content of the document in which it occurs. Such *term weights* are computed from term and document frequency statistics, using mathematical formulae.

Section 2.3.2.1 presents an overview of how IR indices are usually structured to store straight-forward mappings between terms and documents. Section 2.3.2.2 presents an overview of how IR systems compute and store term weights, during indexing. The computational costs of creating, storing, maintaining and processing indices are addressed by the field of IR efficiency, which is presented briefly in Section 2.6.

2.3.2.1 Index data structures

An *index* is a data structure that maps terms to the documents that contain them (Zobel & Moffat, 2006). In IR systems, the use of an index allows for query

[1] *Term* usually refers to a word that has been processed (i.e., stemmed), while *word* usually refers to a word in its grammatically correct form. In IR, *term* and *word* can be used interchangeably, even though the former is more accurate.

processing to be restricted to documents that contain at least one of the query terms. Many different types of index have been described (see Baeza-Yates *et al.* (2002) and Zobel *et al.* (1996) for more information). The most efficient index structure is the *inverted file*[1]. Typically, an inverted file is a collection of lists, one per term, recording the identifiers of the documents containing that term. (A document identifier is usually a document number.) Specifically, an inverted file index consists of two major components:

1. A *vocabulary*

2. A set of *inverted lists*

The vocabulary stores for each distinct term:

- A count of the documents containing the term

- A pointer to the start of the corresponding *inverted list*

The vocabulary may be pre-processed, by stopword removal (Section 2.3.1.1) and stemming (Section 2.3.1.2). In IR systems operating on the Web, any visible component of a Web page might be reasonably used as a query term and hence it is often included in the vocabulary (e.g., numbers, or parts of the URL of Web pages).

In the set of *inverted lists*, each list stores for the corresponding term:

- The identifiers of documents containing the term (also called *postings*)

- The associated set of frequencies of the terms contained in a document

For example, an inverted list can consist of sequences of $< d, f_{t,d} >$ pairs, where d is a document identifier, and $f_{t,d}$ is the frequency of a term in a document. This is an example of a *document level* index, because it indicates whether a term occurs in a document or not, but does not contain information about precisely where the term appears. Alternatively, inverted lists may be augmented with further information, i.e., recording word positions within documents, or co-occurring terms. Such information can be used to enhance the matching of documents to queries (e.g., by considering more significant the terms that co-occur with query terms), or for advanced retrieval options (e.g., using queries

[1] Even though today the inverted file is generally accepted as the most efficient index structure, not all early studies agreed with this, given the resources of the time (Bird *et al.*, 1978; Haskin, 1980; Salton, 1972).

that contain phrases). IR using positional or phrasal information is discussed in Section 2.5.2.

In addition to the inverted file, in a complete IR system, several other indexing structures can also be used, e.g., a table that maps document identifiers to disk locations, or a *direct index* that stores a list of terms that appear in a document for each document.

Finally, when an index is constructed and no information is altered (e.g., added) in it, it is called *static*. On the contrary, when the collection indexed changes over time, e.g., in the case of Web search engines, with data being added to the index, the index is called *dynamic*.

2.3.2.2 Index term weighting

In order to match documents to queries, a standard IR system requires three things: the two indexing data structures described above, namely the vocabulary and list of inverted files, and an array of weighted terms (stored separately), which is described here. The *term weights* produced from the *term weighting* process for indexing are then used to match documents to queries, by computing a score of a document for a query, as described in Section 2.4.2.

The aim of the term weighting process for indexing is to select which terms contribute to the document content and hence should be stored in the system. This selection is realised through term weighting formulae, which assign weights to terms. This section presents a brief overview of the main term weighting formulae used for indexing.

Early studies on automatic term weighting for indexing are based on Zipf's law, which states that the product of term frequency and term rank order (i.e., the frequency of term frequency) is approximately constant (Zipf, 1949). Let f_t be the frequency of terms in a given text, and r_f be their rank order, then Zipf showed that a plot of f_t against r_f yields a roughly hyperbolic curve. More simply, this means that there exists a large number of rare terms, and a small number of very frequent terms. Luhn used the frequency of words in a collection to determine automatically which words were sufficiently significant to characterise documents, and their degree of significance (Luhn, 1958). Specifically, Luhn used Zipf's law to specify two cut-offs of term frequency, an upper and a lower. Terms exceeding the upper cut-off were considered common, and terms below the lower cut-off were considered rare, and therefore not contributing significantly to the content of the document. Luhn assumed that the ability of words to discriminate content reached a peak at a rank order position half way between the two cut-offs, and from the peak fell off in either direction reducing to almost zero at the cut-off points. This provided a simple weighting scheme for the keywords in each document.

Luhn's efforts to automatically assign weights to terms according to their contribution to the document content in which they occur were further exploited (Bookstein & Swanson, 1974; Cooper & Maron, 1978; Damerau, 1965; Harter, 1974; Luhn, 1960; Maron & Kuhns, 1960; Yu & Salton, 1976), and soon extended to include further statistics, apart from the frequency and frequency rank of words. Specifically, weighting formulae were suggested, which processed a small number of fundamental statistical values, such as the frequency of a term in a document/query/collection, the number of documents containing one or more occurrences of a query term, and the number of documents/terms in the collection. Today, these are the basic values typically combined to assign weights to individual terms, which represent the contribution of each term to the document content.

Generally, the basic statistical values described above are combined in a way that results in three monotonicity observations being enforced (Zobel & Moffat, 2006):

1. Less weight is given to terms that occur in many documents, because such words are not likely to indicate the content of any document in particular.

2. More weight is given to terms that occur many times in a document, because such words are likely to indicate the document content.

3. Less weight is given to documents that contain many terms, in order to avoid document length bias.

Hence, most term weighting formulae aim to favour terms that appear to be discriminative, and to reduce the impact of terms that appear to be randomly distributed. An example of how the above statistical values combine is as follows:

- Observation 1, that a term occurring in many documents is not likely to be discriminative, is the *inverse document frequency* (IDF) (Sparck Jones, 1972) of a term, computed as:

$$IDF = \log \frac{N}{f_t} \tag{2.1}$$

  where

  - N is the number of documents in the collection, and
  - f_t is the number of documents in which term t occurs in the collection.

  High IDF means that a term appears in few documents in the collection. The higher the IDF, the higher the discriminatory power of a term in the collection (hence the better).

- Observation 2, that a term occurring many times in a document is likely to indicate the document content, and observation 3, that less weight should be given to longer documents, can be combined in order to compute a 'normalised' term frequency for a term in a document: in IR, term frequency in a document is usually normalised by document length, so that it is computed fairly for long and short documents alike. A simple computation of this normalised term frequency is:

  $$TF = \frac{\log f_{t,d}}{\log dl} \tag{2.2}$$

  where

  - $f_{t,d}$ is the count of term t in document d, and
  - dl is the number of terms in the document (document length).

  High TF means that a term appears frequently in a document. The higher the TF, the more specific the term is with respect to the document (hence the better).

TF and IDF combined make the well-known TF:IDF term weighting formula, which computes the contribution of a word in a document as follows:

$$w_{t,d} = TF \cdot IDF \tag{2.3}$$

where $w_{t,d}$ is the weight of a term in a document.

The contribution of TF:IDF term weighting to IR has been significant. Implicitly or explicitly, most IR approaches contain a TF and IDF component (or their variants) (Robertson, 2004).

Other term weighting formulae used for indexing have also been suggested (for instance the work of Kang & Lee (2005), or see Fuhr (1989) for an earlier overview).

The term weights computed from term weighting are stored by the IR system index, similarly to the way frequency statistics are stored (described in Section 2.3.2.1). Then, the above two components, namely the frequency statistics and the term weights (stored separately), provide all the information required to match documents to queries.

The next section presents how queries and documents are matched by IR systems. Matching documents to queries involves processing the index described in this section.

2.4 Query and document matching

Given a representation of the query and documents, as described in Section 2.3, the IR system matches the document representations to query representations. For simplicity, this is often referred to as matching documents to queries. The aim of this matching is to retrieve documents matched to a query, on the assumption that they are relevant. This section presents the process of matching documents to queries using their representations in the index of the system (Section 2.4.1), and the retrieval models used to realise this matching process (Section 2.4.2).

2.4.1 Matching process

Matching documents to a query can be realised in different ways. One way, called *exhaustive matching*, compares each document in turn to the query, until all documents in the collection are 'exhausted'. The score of each document with respect to the query ($S_{q,d}$) is computed. Then, the best matches (i.e., the documents with the highest $S_{q,d}$ scores) are returned to the user. An example of an algorithm for exhaustive matching between documents and a query is shown in Algorithm 1.

The drawback of exhaustive matching is that every document is explicitly considered, even though typically the number of documents in the collection that are relevant to a query is only a tiny fraction of the total number of documents in the collection (Zobel & Moffat, 2006). Hence, with exhaustive matching, for most documents, the vast majority of matching values are insignificant. Exhaustive matching is suitable only when the collection is small or is highly relative to the query rate.

Algorithm 1 Exhaustive matching of documents to a query

```
for each term $t$ in query $q$ do
    compute $w_{t,q}$
end for
for each document $d$ in the collection do
    set $S_{q,d} \leftarrow 0$
    for each query term $t$ do
        calculate $w_{t,d}$
        set $S_{q,d} \leftarrow S_{q,d} + w_{t,q} \cdot w_{t,d}$
    end for
    calculate $W_d$
    set $S_{q,d} \leftarrow S_{q,d}/W_d$
end for
identify the $r$ greatest $S_{q,d}$ values and return the corresponding documents.
```

Another way for matching documents to a query, called *indexed matching*, uses the inverted lists of the IR system index. (Inverted lists and other index structures were discussed in Section 2.3.2.1.) With indexed matching, query terms are processed one at a time. This is represented by creating an array of matching scores referred to as *accumulators*, one for each document. Initially, each document has a match zero to the query. Then, for each query term, the accumulator for each document mentioned in the given term's inverted list is increased by the contribution of the term to the matching score of the document for the query. Once all query terms are processed, matching scores $S_{q,d}$ are calculated. These scores represent how much a document representation matches a query representation. Finally, the documents with the highest matching scores are returned to the user. An example of an algorithm for indexed matching between a document and a query is shown in Algorithm 2.

Indexed matching is computationally more economical than exhaustive matching, because it processes only documents that contain query terms, not all the documents in the collection.

Algorithm 2 Indexed matching of documents to a query

```
for each document $d$ do
    allocate an accumulator $A_d$
    set $A_d \leftarrow 0$
    for each term $t$ in query $q$ do
        calculate $w_{t,q}$
        fetch the inverted list for $t$
        for each pair $< d, f_{t,d} >$ in the inverted list do
            calculate $w_{t,d}$
            set $A_d \leftarrow A_d + w_{t,q} \cdot w_{t,d}$
        end for
    end for
    read the array of $W_d$ values
    for each $A_d > 0$ do
        set $S_{q,d} \leftarrow \frac{A_d}{W_d}$
    end for
end for
identify the $r$ greatest $S_{q,d}$ values and return the corresponding documents.
```

The next section presents the main models used to match documents to queries.

2.4.2 Matching models

The process of matching documents to queries, presented in Section 2.4.1, can be realised by different matching models. An overview of such matching models is presented in this section.

The common underlying principle of most matching models (Boolean models excepted) is that the better the match between a document representation and a query representation, the greater the likelihood that a user will find the document relevant to the query. Generally, matching models are separated into different categories, the main of which are: *Boolean*, *Vector Space*, *Probabilistic*, and *Language Models*[1]. (Note that language models can also be seen as a branch of probabilistic models, according to Lafferty & Zhai (2003).) Other categories of matching models have also been proposed in the past, for example

[1] Language models are a general category of statistical models, which are briefly introduced in Chapter 4. The type of language models referred to here are language models for IR, and they are related but not identical to the general class of statistical language models. The difference between language models for IR and statistical language models is presented in Section 4.2.2.

using logic (van Rijsbergen, 1986), or inference networks (Turtle & Croft, 1991). Surveys and overviews of retrieval models in general can be found in Salton & Buckley (1988); Zobel & Moffat (2006); Zobel *et al.* (1998).

The remainder of this section presents some of the main categories of matching models, with emphasis on the models used in this thesis (for the experiments described in Chapter 7).

2.4.2.1 Boolean models

In order to match documents to queries, Boolean models (Fox *et al.*, 1992) use Boolean logic and set theory, and treat the query and the documents as sets of terms. The user needs to formulate the query as a boolean statement. An example of a boolean query is `information` **AND** `search` **AND** (**NOT** `storage`). The Boolean model for this query would retrieve all the documents containing the terms `information, search` and not containing the term `storage`.

Boolean models are exact match models: documents are retrieved only if there is an exact match between the document terms and query terms. The matched documents are then presented to the user as a set, without any particular ranking.

Boolean models are the earliest form of matching models. The restrictions in query formulation and lack of ranking of the results have been criticised for the Boolean models (Salton & McGill, 1983). As a result, today research on Boolean models is increasingly of historical interest (Zobel & Moffat, 2006).

2.4.2.2 Vector space models

Vector space models, formally explained by Salton *et al.* (1975), but in use much earlier (Ivie, 1966; Salton, 1962), represent terms as vectors in a multi-dimensional linear space: in order to match documents to queries, queries and documents are represented as vectors of index terms. There is one component in each vector for every distinct term that occurs in the collection. The vector for a document is of size n and contains an entry for each distinct term (where n is the number of terms in the document). The components in the vector are filled with weights that are computed for each term in the collection. These weights are computed using the term weighting formulae described in Section 2.3.2.2, so that the more often the term appears in the document and the less often it appears in all other documents, the higher the weight. Similarly, a vector is constructed for the terms found in the query.

Once the vectors are constructed, the distance between the query and document vectors, or the size of the angle between the vectors, is used to compute a score of match between a document and a query ($S_{q,d}$). The closer the query vector is to the document vector, the more relevant the document is assumed to be to the query. For vector space models, $S_{q,d}$ is often referred to as similarity coefficient between the query and document vectors.

Several different measures for computing the distance between the document and query vectors have been suggested (van Rijsbergen, 1979). A typical formulation calculates the cosine of the angle in n-dimensional space between a query vector $< w_{t,q} >$ and a document vector $< w_{t,d} >$ as follows:

$$S_{q,d} = \frac{\sum_{t \in q} w_{t,q} \cdot w_{t,d}}{W_q \cdot W_d} \tag{2.4}$$

where

- $S_{q,d}$ is the score of match between a document d and a query q;
- $w_{t,q}$ is the weight of a query term, computed with a term weighting model as the ones described in Section 2.3.2.2. For example, $w_{t,q}$ can be computed by varying the IDF formula given in Equation 2.1, page 16, as follows:

  $$w_{t,q} = \log(1 + \frac{N}{f_t}) \tag{2.5}$$

  where

  - N is the number of documents in the collection, and
  - f_t is the number of documents in which term t occurs in the collection.

- $w_{t,d}$ is the weight of a document term, also computed with a term weighting formula as the ones described in Section 2.3.2.2. For example, $w_{t,d}$ can be computed by varying the TF formula given in Equation 2.2, page 17, as follows:

  $$w_{t,d} = 1 + \log f_{t,d} \tag{2.6}$$

  where $f_{t,d}$ is the frequency of term t in document d;

- W_d is the weight of a document, computed from the weights of the terms it contains. For example,

  $$W_d = \sqrt{\sum_{t \in q} w_{t,d}^2} \tag{2.7}$$

- W_q is the weight of a query, computed from the weights of the terms it contains. For example,

$$W_q = \sqrt{\sum_{t \in q} w_{t,q}^2} \tag{2.8}$$

Another way for computing the similarity between a query and a document vector is the Euclidean distance between the two vectors, which is used in the experiments described in Chapter 7. The Euclidean distance formulation for computing the similarity between a query and a document vector is:

$$S_{q,d} = \sum_{t \in q} w_{t,q} \cdot w_{t,d} \tag{2.9}$$

where $w_{t,q}$ and $w_{t,d}$ can be computed using Equations 2.5 and 2.6, respectively.

A significant variation of the vector space model formulation for matching documents to queries has been the introduction of document length pivoting by Singhal *et al.* (1996), which addressed the issue that document length and likelihood of relevance are correlated.

Vector space models are best match models: they retrieve documents that best match the query. This means that the retrieved documents may contain some, but not all, of the query terms. Best match models typically rank documents according to their relevance to the query.

Today, vector space models have become standard for matching documents to queries. For much of the history of IR, the principal alternative to vector space models has been probabilistic models (Zobel & Moffat, 2006). Probabilistic models are presented next.

2.4.2.3 Probabilistic models

In order to match documents to queries, probabilistic models estimate the probability of relevance for a document given a query. The basis of probabilistic models is the *probability ranking principle*, explored by Maron & Kuhns (1960), and later formalised by Robertson (1977). Given a collection of documents and a query, the probability ranking principle views the documents in the collection as belonging to either a relevant class with respect to a query, or to a non-relevant class with respect to a query. Then, the probability ranking principle suggests that, for optimum retrieval performance, the retrieved documents should be ranked by their odds of being observed in the relevant class:

$$\frac{P(d|r)}{P(d|n)} \tag{2.10}$$

where

- d is a document,
- r is the relevant class of documents,
- n is the non-relevant class of documents, and
- $P(d|r)$ (resp. $P(d|n)$) is the probability of observing document d in the relevant (resp. non-relevant) class of documents.

Hence, the probability ranking principle, and by extension probabilistic models, assume that there is some knowledge of the distribution of terms in the relevant documents. Similarly to vector space models, probabilistic models are best match models: they retrieve documents that best match the query, and rank documents according to their relevance to the query.

One of the most popular probabilistic models, introduced by Robertson & Sparck Jones (1976), ranks documents by the probability of belonging to the relevant class of documents for a query based on the estimated word occurrence characteristics of those classes. Today, among the most popular probabilistic models are the well-known **BM25** model from the Best Match (BM) family of models used in the Okapi system (Robertson & Walker, 1994), and the **PL2** model from the Divergence From Randomness (DFR) framework (Amati, 2003). Many of the underpinnings of probabilistic IR are summarised by Sparck Jones *et al.* (2000), who give a detailed derivation of BM25. BM25 and PL2, which are used in the experiments described in Chapter 7, are described separately next.

2.4.2.3.1 Best Match 25 (BM25) BM25 computes the matching score $S_{q,d}$ between a document d and a query q as follows:

$$S_{q,d} = \sum_{t \in q} w_{t,q} \cdot w_{t,d} \tag{2.11}$$

where

- $w_{t,q}$ is the weight of a query term, given by:

$$w_{t,q} = \log(\frac{N - f_t + 0.5}{f_t + 0.5}) \cdot \frac{(k_3 + 1) \cdot f_{t,q}}{k_3 + f_{t,q}} \tag{2.12}$$

  where

- N is the number of documents in the collection;
- f_t is the frequency of documents containing term t in the collection;
- k_3 is a parameter, the recommended value of which is 1000 (Robertson & Walker, 1994); and
- $f_{t,q}$ is the term frequency in the query.

- $w_{t,d}$ is the weight of a document term, given by

$$w_{t,d} = \frac{(k_1 + 1) \cdot f_{t,d}}{K + f_{t,d}} \tag{2.13}$$

where

- k_1 is a parameter, the recommended value of which is 1.2 (Robertson & Walker, 1994);
- $f_{t,d}$ is the term frequency in the document; and
- K is given by:

$$K = k_1((1 - b) + b \cdot \frac{dl}{avdl}) \tag{2.14}$$

where

  - b is a parameter, the recommended value of which is 0.75 (Robertson & Walker, 1994);
  - dl is the document length, measured in any suitable units (e.g., indexed terms, bytes, and so on); and
  - $avdl$ is the average document length in the collection, measured similarly to dl.

Combining the above, BM25 computes the matching score of a document d for a query q as follows:

$$S_{q,d} = \sum_{t \in q} \log(\frac{N - f_t + 0.5}{f_t + 0.5}) \cdot \frac{(k_3 + 1) \cdot f_{t,q}}{k_3 + f_{t,q}} \cdot \frac{(k_1 + 1) \cdot f_{t,d}}{K + f_{t,d}} \tag{2.15}$$

2.4.2.3.2 Poisson Laplace 2 (PL2) The PL2 matching model belongs to the DFR framework of models. DFR models comprise three components: a *randomness model* (RM), an *information gain model* (GM), and a *term frequency normalisation* model (TFN). The randomness model estimates the probability that a term occurs in a document randomly. The less randomly a term occurs in a document, the more information it conveys. The information gain model estimates the probability that a term is a good descriptor of a document, within a collection. The information gain model estimates the informative content risk $1 - P$ of the probability $Prisk$ that a term t is a good descriptor for a document. Good descriptors are terms occurring rarely in the collection, but frequently in the subset of documents relevant to the query (this notion was presented in Section 2.3). The term frequency normalisation model adjusts the frequency of a term in a document, on the basis of the length of that document and the average document length in the whole collection, so that longer documents do not have an unfair advantage over shorter documents.

PL2 computes the matching score $S_{q,d}$ between a document d to a query q as follows:

$$S_{q,d} = \sum_{t \in q} w_{t,q} \cdot w_{t,d} \tag{2.16}$$

where

- $w_{t,q}$ is the query term weight, given by:

  $$w_{t,q} = \frac{f_{t,q}}{fmax_{t,q}} \tag{2.17}$$

  where

  - $f_{t,q}$ is the query term frequency; and
  - $fmax_{t,q}$ is the maximum query term frequency.

- $w_{t,d}$ is the weight of the term t in document d, given by:

  $$w_{t,d} = (1 - P_{risk}) \cdot (-log_2 P_{RM}) \tag{2.18}$$

  where

  - the information gain component is computed from P_{risk}, which is the conditional probability of having one more occurrence of a term in a document, where the term appears $f_{t,d}$ times already. P_{risk} is computed as follows:

    $$1 - P_{risk} = 1 - \frac{f_{t,d}}{1 + f_{t,d}} \tag{2.19}$$

    $$= \frac{1}{1 + f_{t,d}} \tag{2.20}$$

    where $f_{t,d}$ is the term frequency in document d;

- the randomness model component is computed as follows:

$$-\log_2 P_{RM} = f_{t,d} \cdot \log_2 \frac{f_{t,d}}{\lambda} + (\lambda - f_{t,d}) \cdot \log_2 e + 0.5 \cdot \log_2 (2\pi \cdot f_{t,d}) \quad (2.21)$$

  where $\lambda = \frac{f_{t,c}}{C}$, where

  - $f_{t,c}$ is the term frequency in the collection,
  - C is the number of all terms in the collection.

- the term frequency normalisation (tfn) component is computed as follows:

$$tfn = f_{t,d} \cdot \log_2 (1 + c \cdot \frac{avdl}{dl}) \quad (2.22)$$

  where

  - c is a parameter, the recommended value of which is 7.0 (Amati, 2003);
  - dl is the document length, measured in any suitable units (e.g., indexed terms, bytes, and so on); and
  - $avdl$ is the average document length in the collection, measured similarly to dl.

Combining the above, and replacing $f_{t,d}$ with tfn, PL2 computes the matching score of a document for a query as follows:

$$S_{q,d} = \sum_{t \in q} w_{t,q} \cdot \frac{1}{tfn + 1} (tfn \cdot \log_2 \frac{tfn}{\lambda} \cdot \log_2 e + 0.5 \cdot \log_2 (2\pi \cdot tfn)) \quad (2.23)$$

2.4.2.4 Language models

In order to match documents to queries, language models estimate the probability that a query is generated from a document by generating a ‘language model’ for each document (Ponte & Croft, 1998). The language model of a document is a ‘data model’ consisting of the words occurring in the document. For a document, the probability of occurrence of a query term can be easily estimated (e.g., using maximum likelihood (Ng, 1999)). Then, for a given query, the documents are ranked according to the probability that the data model of the corresponding document generates the query. This type of estimation is usually smoothed to avoid assigning zero probabilities to terms not occurring in a document. Several smoothing techniques have been suggested (see Zhai & Lafferty (2001) for an overview).

A language model computes the matching score $S_{q,d}$ of a document d for a query q as follows:

$$S_{q,d} = \sum_{t \in q} w_{t,q} \cdot w_{t,d} \tag{2.24}$$

where

- $w_{t,q}$ is the query term weight, given by the frequency of a term in the query ($w_{t,q} = f_{t,q}$), and
- $w_{t,d}$ is the weight of a term in document d, given by:

  $$w_{t,d} \overset{rank}{=} \prod_{t \in q} P(q|d) \tag{2.25}$$

  where $P(q|d)$ is the likelihood of the query according to the document model. In the language modelling formalism, given a query $q = \{q_1, q_2, ..., q_n\}$, where $\{q_1, q_2, ..., q_n\}$ is the sequence of query terms, $P(q|d)$ is computed as follows:

  $$P(q|d) = \prod_i P(q_i|q_1, q_2, ..., q_{i-1}, d) \tag{2.26}$$

$P(q|d)$ can be computed not only for each individual term separately, but also for contiguous sequences of terms. When processing sequences of n terms, language models are called n-gram models. Typically, for 1-gram language models, Equation 2.26 is approximated as follows:

$$P(q|d) \approx \prod_i P(q_i|d) \tag{2.27}$$

As mentioned above, there are several alternatives of computing $P(q|d)$, so that the overall estimation is not affected negatively if query terms occur very rarely or not at all in the collection. One such alternative, which uses Dirichlet smoothing, is given by:

$$P(q|d) = Pmle(q|d) \cdot \frac{dl}{dl + \mu} + P(q|C) \cdot \frac{\mu}{dl + \mu} \tag{2.28}$$

where

- $Pmle(q|d)$ is the maximum likelihood of a query term occurring in a document, given by:

  $$Pmle(q|d) = \frac{f_{t,d}}{dl} \tag{2.29}$$

  where

  - $f_{t,d}$ is the frequency of a term in a document, and
  - dl is the document length, measured in any suitable units (e.g., indexed terms, bytes, and so on).

- $P(q|C)$ is the probability of a query term occurring in a collection, given by:

$$Pmle(q|C) = \frac{f_{t,c}}{C} \tag{2.30}$$

  where

  - $f_{t,c}$ is the frequency of a term in the whole collection, and
  - C is the number of all terms in the collection, and

- μ is a parameter, typically set to 2500 (Zhai & Lafferty, 2001).

Another alternative for computing $P(q|d)$, which uses Jelinek-Mercer (JM) smoothing, is given by:

$$P(q|d) = (1 - \lambda) \cdot Pmle(q|d) + \lambda \cdot P(q|C) \tag{2.31}$$

where

- $Pmle(q|d)$ is as defined in Equation 2.29;
- $P(q|C)$ is as defined in Equation 2.30; and
- λ is a parameter, which should be set between 0-1 (Zhai & Lafferty, 2001).

In language models, in addition to estimating the likelihood of the query having been generated from a document, several other alternatives have been proposed: one alternative, proposed by Lavrenko & Croft (2001), is to associate a language model with the query or topic of interest, and to rank documents based on the probability of them being generated by the query language model (i.e., the *document-likelihood*); another alternative, proposed by Lafferty & Zhai (2001), is a *query document model similarity* based on the *Risk Minimization Framework*, according to which documents are ranked based on the similarity between language models associated with the query and a document.

Initially, n-gram language models of $n > 1$, in particular smoothed n-gram language models, were shown to perform better than smoothed 1-gram language models (Miller *et al.*, 1999; Song & Croft, 1999). More recently, 1-gram language models have been reported to outperform language models of $n > 1$ (Zobel & Moffat, 2006).

It has been shown that 1-gram language models can be estimated using the settings of a vector space or probabilistic model (Hiemstra, 2000). Similarly to vector space and probabilistic models, language models are best match models: they retrieve documents that best match the query, and rank documents according to their relevance to the query.

It has been argued that probabilistic and language models are equivalent from a probabilistic point of view, but differ in terms of statistical estimation: probabilistic models estimate a model for relevant documents based on a query, while language models estimate a model for relevant queries based on a document (Lafferty & Zhai, 2003).

2.5 Retrieval boosting techniques

Sections 2.3 and 2.4 have presented basic concepts of IR systems, i.e., the minimum resources and processing required to have an operational IR system. This section presents two performance enhancing techniques, which are often used on top of basic IR operations, in order to boost IR system performance. Specifically, these performance boosting techniques are:

- Feeding back to the system evidence about what is considered relevant to the query (*relevance feedback*, presented in Section 2.5.1).
- Considering phrases or terms co-occurring often, when matching documents to queries (*phrasal IR*, presented in Section 2.5.2).

2.5.1 Relevance feedback

The aim of *relevance feedback* is to enrich queries with more relevant words in order to facilitate the retrieval of more relevant documents (Zhang *et al.*, 2004). This can be useful when the initial query contains few or non-informative terms, which makes the task of retrieving relevant documents harder (Carmel *et al.*, 2006; van Rijsbergen *et al.*, 1981). The main methodology of relevance feedback is firstly to expand the initial user query with relevant words, and secondly to re-submit the expanded query to the IR system.

Relevance feedback has been widely explored in IR. In *explicit relevance feedback* systems, users identify the answers that are of value (and perhaps others that are not) to their query, and this information is incorporated into a revised query, which is re-submitted to the system in order to facilitate the retrieval of relevant documents (Harman, 1988). Much of this research assumes that queries submitted by the same user are independent of each other. A problem with explicit relevance feedback is that in real-life IR systems, it has not been proven easy to gather relevance feedback from users.

An alternative to explicit relevance feedback is *implicit relevance feedback*. Implicit relevance feedback systems incorporate information about which answers are likely to be relevant into revised queries, having collected this information without asking the user explicitly, but instead by recording some elements of user behaviour implicitly. For example, in a system processing large numbers of queries, is it not hard to identify which answers users choose to view. The action of a user clicking on a link (referred to as *click-through*) can be interpreted as a vote for a document. Such a voting can be used to alter the weights computed when matching documents to queries, and thus the document ordering in the ranking generated for subsequent queries, even if the subsequent queries differ from the one that triggered the click-through.

An alternative to relevance feedback from the user (either explicitly, or implicitly) is *pseudo-relevance feedback* (PRF). PRF is an automatic technique that does not involve the user. In PRF, after documents have been matched to queries, the query is expanded with terms that are correlated with relevance (Robertson, 1990), computed using term weighting formulae such as the ones presented in Section 2.3.2.2. The expanded query is re-weighted and re-matched to documents (Buckley *et al.*, 1994). Amati (2003); Rocchio (1971); Salton & Buckley (1980); Xu & Croft (1996) describe various PRF models that enhance retrieval performance.

PRF is criticised for involving some heuristic tuning, for instance in deciding how many assumed relevant terms to add to the query. These heuristics can be collection-dependent, or also affected by the initial query length (Carpineto *et al.*, 2001). Tuning PRF separately for different collections or query length is neither a feasible nor robust option. Recent work on query prediction, i.e., predicting the difficulty of a query in retrieving relevant documents, has allowed for PRF to be applied selectively only to those queries that retrieve relevant documents (Carmel *et al.*, 2006).

2.5.2 Phrases and co-occurring terms

When matching documents to queries, most retrieval models process single words individually and regardless of their context, hence they are called *bag of words* models. By doing so, they assume that terms occur independently of each other (*term independence assumption*). However, this is more a matter of mathematical convenience rather than a reality in natural language (Kilgarriff, 2005; Nallapati & Allan, 2002). In IR, efforts to process words in the contexts in which they occur are often referred to as modelling *dependence, co-occurrence, adjacency* and *lexical affinities*[1], or collectivelly as *phrasal IR*.

Broadly speaking, efforts to model term co-occurrence and term dependence in IR typically aim to model phrases, found in queries and/or documents. Modelling phrases in queries is motivated by the fact that a small but significant fraction of user queries include an explicit phrase (Zobel & Moffat, 2006). Modelling phrases in documents is motivated by the intuition to consider more relevant documents in which terms appear in the same order and patterns as they appear in the query, and less relevant documents in which the terms are separated (Smith & Devine, 1985).

Generally in phrasal IR, phrases are detected using either statistical or linguistic information. Research on the general topic of phrasal IR began with the early work on statistical term associations (Doyle, 1962; Giuliano & Jones, 1963; Lesk, 1969; Stiles, 1961) and syntax-based approaches (Baxendale, 1958; Earl, 1972; Salton, 1966). Investigation continued with work on probabilistic term dependence models (Harper & van Rijsbergen, 1978; Salton *et al.*, 1983; Turtle & Croft, 1991; van Rijsbergen, 1977; Yu *et al.*, 1983), syntactic methods (Dillon & Gray, 1983; Metzler *et al.*, 1984; Smeaton, 1986) and statistical approaches (Fagan, 1989; Lewis, 1992). More recently, relevant research has focussed on statistical methods, mostly using language modelling (Metzler & Croft, 2005; Mishne & de Rijke, 2005; Nallapati & Allan, 2002; Song & Croft, 1999), but not exclusively (Losee, 1994; Plachouras & Ounis, 2007).

Often, the use of co-occurrence information in IR results in a reduction of retrieval effectiveness (Salton *et al.*, 1983). It has been suggested that this is due to the fact that the term relationships modelled tend to have little discriminating power (Metzler & Croft, 2005) because:

[1] Strictly speaking *dependence, co-occurrence, adjacency* or *lexical affinities* are not synonyms (Heylighen & Dewaele, 2002; Sinclair, 1991), but in IR they have been used intechangeably.

- For phrasal IR that uses probabilistic matching models, term dependence must be estimated in the relevant and the non-relevant classes, where there is often a very small or non-existent sample of relevant/non-relevant documents available to estimate the model parameters from. Hence there is not enough data to make an accurate estimation.

- The document collections used in past work consist of a very small number of short documents, hence there is very little hope of accurately modelling term dependencies when most pairs of terms only occur a handful of times, if at all.

Recently, Metzler & Croft (2005) have reported significant improvement in retrieval perfomance when modelling term dependence.

2.6 Information retrieval system efficiency

Sections 2.3, 2.4 and 2.5 have presented the main resources, processes, and techniques typically used to represent and match documents and queries in an IR system. This section presents briefly the main efficiency concerns relating to these processes.

IR system efficiency is affected by the size of the collections from which information is retrieved. Collection size can vary dramatically. For instance, the text of all books held in a small university library might occupy around 100 gigabytes (GB) of disk space, whereas the complete text of the Web in 2005 was estimated to occupy several tens of terabytes (TB) (Zobel & Moffat, 2006).

In addition, the computational costs of storing and matching documents to queries are significant:

- *Disk space* is required for the index at 20%-60% of the original size of the collection (indices and their data structures were presented in Section 2.3.2.1).

- *Memory* is required for an accumulator for each document and for some or all of the vocabulary (the role of the accumulator in matching documents to queries was described in Section 2.4.1).

- *Central Processing Unit* (CPU) time is required for processing inverted lists and accumulators.

- *Disk traffic* is used to fetch inverted lists.

How to reduce these costs without hurting retrieval effectiveness is one of the aims of the efficiency field of IR.

Efficiency is of outmost importance in operational IR systems. For further information on efficiency for IR, see Frakes & Baeza-Yates (1992); Grossman & Frieder (2004); Witten *et al.* (1999).

Next, two main facets of efficiency are introduced briefly, namely *data compression* (in Section 2.6.1) and *data distribution* (in Section 2.6.2).

2.6.1 Data compression

The information stored in the index of an IR system is usually 'encoded', i.e., stored in a compressed form, so that large collections of documents are efficiently indexed without large computational costs (Witten *et al.*, 1999). Specifically, with appropriate compression techniques, compression has the following advantages:

- The costs of index construction can be reduced
- The disk space consumption needed to store the index can be reduced
- The costs of index maintainance can be reduced
- The disk traffic of the system can be reduced:
  - The overall transfer costs during query-document matching time can be reduced, because the inverted lists of the index are shorter. (Inverted lists were presented in Section 2.3.2.1).
  - The overall seek times can be reduced because the index is smaller.

The principal disadvantage of compression is that inverted lists must be decoded before they are used. A related problem is that they may need to be re-coded if and when they are updated (i.e., in the case of dynamic indices, presented in Section 2.3.2.1), and, for some codes, the addition of new information can require complex decoding. Generally, if the index is larger than the available main memory of the system, there is no disadvantage to compression (Zobel & Moffat, 2006). And even if the index is in memory, processing can be faster than for uncompressed data. The use of appropriate index compression techniques is an important facet of the design of an efficient IR system.

2.6.2 Data distribution

An IR system index can consist of either a *single* or a *distributed* body of data, the latter of which is usually prefered to facilitate the processing of very large collections, which cannot be supported by a single machine. For example, in mid-2004, the Google search engine processed more than 200 million queries a day against more than 20TB of data, using more that 20,000 computers (Zobel & Moffat, 2006). Data distribution refers to the fact that a document collection and its index are split across multiple machines. Hence, answers to a query must be synthesised from the various collection components. Data distribution is often used with *replication* (or *mirroring*). Replication involves making enough identical copies of the system, so that the required query load can be handled. A distributed IR system in which the set of retrieved documents is synthesised from the possibly overlapping sets provided by a range of different services is called a *metasearcher*.

More information on distribution and Web IR can be found in Arusu *et al.* (2001); Kobayashi & Takeda (2000).

2.7 Information retrieval evaluation

In Section 2.2, it was stated that, in an IR system, a document matches the user's information need if the user perceives it to be relevant. However, relevance is not an exact notion: a document that contains some but not all of the query terms might be relevant to the user information need, while a document that contains all of the query terms might be irrelevant to the user need. Most users are aware that only some of the matches returned by the IR system will be relevant to their need, and also that different IR systems may return different matches for the same query (Zobel & Moffat, 2006).

This inexactitude introduces the notion of *effectiveness*: informally, an IR system is effective if a good proportion of the first matched documents returned are relevant. Formally, given a collection C, the primary goal of an IR system is to identify a set $C_R \subset C$ of documents relevant to a query. This is a set-based decision task, but, in practice, most retrieval systems are evaluated by how well they *rank* the documents in the collection (van Rijsbergen, 1979). Let $d_1, d_2, \ldots d_N$ denote some ordering of the documents in the collection. Then, for every rank k, *recall* is the number of relevant documents that were observed in the set $\{d_1 \ldots d_k\}$, divided by the total number of relevant documents in the collection. Similarly, *precision* is the number of relevant documents among $\{d_1 \ldots d_k\}$, divided by k. System performance is evaluated by comparing precision at different levels of recall. A common objective is to increase precision at all levels of recall. For applications that require interaction with a user, it is common to report precision at specific ranks, e.g., after 5 or 10 retrieved documents.

When one desires a single number as a measure of performance, several alternatives have been proposed. A popular choice is the *average precision*, defined as the arithmetic average of precision at every rank where the relevant document occurs, using zero as the precision for relevant documents that are not retrieved. Geometrically, average precision is the equivalent to the area underneath an uninterpolated recall-precision graph (Buckley & Voorhees, 2004). Another possible choice is *R-precision*, precision that is achieved at rank R, where R is the number of relevant documents in the dataset. In these measures, precision values are usually averaged across a large set of queries with known relevant sets. An example is the *mean average precision* (MAP) measure. An overview of evaluation measures for IR systems is found in Baeza-Yates & Ribeiro-Neto (1999); Demartini & Mizzaro (2006).

IR systems are often evaluated on standard datasets of documents and queries, for which relevant documents are known previously. This evaluation paradigm is part of the Text REtrieval Conference (TREC[1]), an organised effort to support the evaluation of IR methodologies. Overviews and discussions of TREC can be found in Blair (2002), as well as on the annual overviews of the conference proceedings, available on the TREC Website. The idea behind TREC is to evaluate IR systems on standard and controlled datasets. These datasets consist of a collection of documents, with an associated set of queries, along with human relevance judgments about which documents are relevant to the queries. These judgments are not exhaustive, but *pooled*. This means that, for each query, human annotators do not judge all documents in the collection, but only top-ranked documents from a set of retrieval systems. TREC queries usually contain a *title*, *description*, and *narrative* portion. The title contains few keywords; the description includes a brief description of the information need; the narrative contains a longer description of the information need.

In addition to TREC, there are other evaluation paradigms, which are specialised on specific IR branches: for instance, the Cross Language Evaluation Forum (CLEF[2]) addresses mainly monolingual, multilingual, and crosslingual

[1] http://trec.nist.gov/

[2] http:www.clef-campaign.org

IR in European languages; the NII Test Collections for IR Systems (NTCIR[1]) project focusses on monolingual, multilingual, and crosslingual IR in East Asian languages. Both CLEF and NTCIR have been developed in accordance to the TREC paradigm.

Finally, the evaluation of IR systems can be extended to other issues than retrieval effectiveness, namely the efficiency of the system (presented in Section 2.6). Typical criteria for rating IR system efficiency are:

- Indexing speed: how many documents per hour does the system index for a certain distribution over document size?
- Retrieval speed: what is the system's latency as a function of index size?
- Collection cost: how large is the collection stored in the system, in terms of number of documents, or the collection having information distributed across a broad range of topics?

2.8 Summary

This chapter introduced the basic concepts involved in IR systems (Section 2.2), and the main processes for representing documents and queries (Section 2.3), and for matching documents to queries (Section 2.4). In addition, two main retrieval enhancing techniques often used by IR systems were presented (Section 2.5). Issues of IR system efficiency were briefly addressed (Section 2.6). Finally, the main concepts and paradigms of IR evaluation were presented (Section 2.7).

Chapter 3 presents the basic concepts of parts of speech, which are the shallow grammatical categories modelled as n-grams in this thesis.

[1] http:research.nii.ac.jp/ntcir/data/data-en.html

Chapter 3

Basic concepts of parts of speech

3.1 Introduction

This chapter presents parts of speech. First, parts of speech are defined and their main properties are introduced. Then, applications are presented for automatically assigning parts of speech to words (*POS tagging*), as well as IR applications using parts of speech to enhance retrieval performance.

This chapter is organised as follows: Section 3.2 introduces parts of speech. Section 3.3 presents a linguistic theory for ranking parts of speech, which is used extensively in this thesis. Section 3.4 presents automatic ways for assigning parts of speech to words, and in particular three standard POS taggers. Section 3.5 gives an overview of IR applications that use parts of speech to enhance retrieval performance. Section 3.6 summarises and concludes this chapter.

3.2 Part of speech categories

Parts of speech are grammatical categories of words, such as noun or verb. There are several levels of grammatical categories in language. Using the terminology of Lyons (1977), the parts of speech are *primary grammatical categories*. *Secondary grammatical categories* are such notions as verbal tense, or nominal case, which relate to the inflection or conjugation of words. *Functional grammatical categories* are the traditional syntactic notions of subject, predicate, object, and so on, which relate to the discourse roles of words (Hopper & Thompson, 1984).

Parts of speech are found with some variation in most languages, and are much fewer in number than the number of words in language. Also, unlike words, there exists a specific number of parts of speech, which can range according to different classifications. A word can have more than one part of speech, e.g., `book` can be a noun (`to read a book`) and a verb (`to book a flight`). An introduction into parts of speech can be found in Radford (1988).

Categorising words into parts of speech can be traced to 4th century BC studies of Sanskrit and ancient Greek (Lyons, 1977, page 19), which independently observed that language and mathematics alike are made up of individual units, defined and arranged through rules. Some of these units operate as constants, while other units operate as variables. Two classes were defined as universal and necessary categories of language: (i) nouns, and (ii) verbs[1] and adjectives. These grammatical classes were defined on logical grounds: nouns were the subject of a predication (the thing about which something is said), and verbs and adjectives expressed the action or quality predicated. All other language units received little attention. Thus, language was formally defined as a structure-based setting, where specific and indispensable units interacted with circumstantial and dispenable units (Lyons, 1977, Chapter 1.2).

Today grammatical categories are extended to include more parts of speech, such as prepositions, pronouns, and so on. A widely accepted part of speech categorisation is the Penn TreeBank set (Marcus *et al.*, 1993), the primary grammatical categories of which are shown in Table 3.1. The distinction of modern categories from the early 'fundamental' categories remains (Crystal, 1967): modern parts of speech are separated into *major* (nouns, verbs, participles[2], adjectives, and sometimes adverbs[3]) and *minor* (all other categories). Major categories are also known as *open*, because membership to this class is open to new nouns, verbs, and adjectives that are formed in language. Minor categories are also known as *closed*, because membership to this class is mostly fixed: for example, new prepositions are not coined.

This bifurcation of primary grammatical categories into two classes is widely accepted by theorists and practitioners of linguistics and language technologies alike. Linguists often compare it to the Aristotelian opposition of 'matter' and 'form': the open class parts of speech 'signify' the objects of thought which constitute the 'matter' of discourse; the closed parts of speech do not 'signify' anything of themselves, but merely contribute to the total meaning of sentences,

[1] The category of verbs included participles.

[2] Even today, participles are sometimes classified as verbs, and sometimes classified separately. When participles are classified separately, they share the properties of verbs. In this thesis, participles are classified separately than verbs, following the Penn TreeBank classification.

[3] The classification of adverbs has always been borderline (Lyons, 1968).

Penn Treebank classification: primary parts of speech			
part of speech	**abbr.**	**part of speech**	**abbr.**
adjective	JJ	participle	VR
adverb	RB	particle	RP
conjunction	CC	possessive ending	PO
determiner	DT	preposition	IN
modal verb	MD	pronoun	PP
noun	NN	symbol	SY
numeral	CD	verb	VB

Table 3.1: Primary part of speech categories (Penn Treebank set).

by imposing upon them a certain ‘form’, or organisation (Bas *et al.*, 2004, pages 29-64), (Hjelmslev, 1943). This separation is also reminiscent of the distinction traditionally drawn between ‘full’ and ‘empty’ words in Chinese grammatical theory (Lyons, 1977, page 273). In language processing technologies, closed class parts of speech tend to correspond to the stopwords that are often excluded from processing because of their negligible contribution to the overall content (see Section 2.3 for stopword removal in IR, or Mani (2001) for stopword removal in automatic summarisation).

The open and closed class division of parts of speech can be represented using set theoretical notation as follows.

Notation Standard set theoretic notation is used, where $\{...\}$ is an unordered set, and $[...]$ is an ordered set. Let $\{pos\}$ be the set of all parts of speech. Let $\{pos_o\}$ be the set of all open class parts of speech, and $\{pos_c\}$ be the set of all closed class parts of speech.

Open class The set of open class parts of speech $\{pos_o\}$ is a *proper subset* of the set of all parts of speech $\{pos\}$.

$$\{pos_o\} \subset \{pos\} \tag{3.1}$$

This means two things:

- $\{pos_o\} \subseteq \{pos\}$: all members of $\{pos_o\}$ also belong to $\{pos\}$
- $\{pos_o\} \neq \{pos\}$: $\{pos\}$ has at least one member that does not belong to $\{pos_o\}$

Closed class The set of closed class parts of speech $\{pos_c\}$ is a proper subset of the set of all parts of speech $\{pos\}$.

$$\{pos_c\} \subset \{pos\} \tag{3.2}$$

($\{pos_c\} \subseteq \{pos\}$ and $\{pos_c\} \neq \{pos\}$).

All classes The set of all parts of speech $\{pos\}$ is the *union* of the set of open class parts of speech $\{pos_o\}$ and the set of closed class parts of speech $\{pos_c\}$.

$$\{pos\} = \{pos_o\} \cup \{pos_c\} \tag{3.3}$$

This notation is used extensively in the rest of the thesis, and particularly in Section 5.3.

3.3 Jespersen's rank theory

Grouping all major or open parts of speech into one class does not imply that the historical distinction between nouns as the fundamental grammatical unit, versus verbs and adjectives is lost (Ross, 1973). An early formulation of this distinction is Jespersen's *Rank Theory* (Jespersen, 1913, 1929). Jespersen suggested that grammatical categories are semantically definable and subject to ranking. He identified *degrees* of parts of speech:

- *First degree* (or *primary*) parts of speech: nouns.
- *Second degree* (or *secondary*) parts of speech: verbs (including participles) and adjectives.
- *Third degree* (or *tertiary*) parts of speech: adverbs.

Jespersen defined the notion of *degree* in terms of the combinatorial properties of the parts of speech: each part of speech is modified by a part of speech of higher degree. E.g., nouns are modified by verbs, and verbs are modified by adverbs. No more than three degrees are required, because there is no major part of speech with the function to modify parts of speech of the third degree.

Jespersen's ranking may be seen as crude because it does not distinguish between different ranks of open or closed class parts of speech, some of which are likely to be more informative than others. For example, whereas Jespersen ranks all closed class parts of speech in one group, some closed class parts of speech, such as cardinal numbers or modal verbs for instance, may be more informative than other closed class parts of speech, such as determiners or conjunctions for instance. Similarly for open class parts of speech, Jespersen's ranking does not distinguish between different classes of open class parts of speech, some of which are likely to hold more content than others, such as action verbs, emotion verbs, or mass nouns, proper nouns, and so on. Despite this criticism, Jespersen's Rank Theory has influenced linguistics extensively (see Anderson (1997); Newmeyer (2000); O'Grady (1988) for more information) and partially forms the basis of later well-known grammatical theories, such as Categorial Grammars (Chomsky, 1961). This thesis draws from Jespersen's Rank Theory, and not from its derivatives, because of its antecedence and clarity.

Extending the set theoretical notation introduced in Section 3.2, Jespersen's ranking of parts of speech can be formally represented as follows:

Notation Let $\{pos'\}$ be the set of first degree parts of speech, $\{pos''\}$ be the set of second degree parts of speech, and $\{pos'''\}$ be the set of third degree parts of speech.

1st Degree First degree parts of speech are a proper subset of the open class.

$$\{pos'\} \subset \{pos_o\} \tag{3.4}$$

($\{pos'\} \subseteq \{pos_o\}$ and $\{pos'\} \neq \{pos_o\}$).

2nd Degree Second degree parts of speech are a proper subset of the open class.

$$\{pos''\} \subset \{pos_o\} \tag{3.5}$$

($\{pos''\} \subseteq \{pos_o\}$ and $\{pos''\} \neq \{pos_o\}$).

3rd Degree Third degree parts of speech are a proper subset of the closed class.

$$\{pos'''\} \subset \{pos_c\} \tag{3.6}$$

($\{pos'''\} \subseteq \{pos_c\}$ and $\{pos'''\} \neq \{pos_c\}$).

Open Class The open class is the union of first and second degree parts of speech.

$$\{pos_o\} = \{pos'\} \cup \{pos''\} \tag{3.7}$$

This notation is used extensively in the rest of the thesis, and particularly in Section 5.3.

3.4 Part of speech tagging

This section presents briefly the automatic process of annotating previously unseen sequences of words forming phrases or sentences with their respective POS tags. This process is often called *part of speech (POS) tagging*, but may also be referred to as POS disambiguation. Given a sequence of words, the task of a POS tagger is to make a decision about the part of speech of each word. Even though this decision cannot be made without understanding the meaning of the input sequence of words, it can be reasonably approximated using automatic means that ignore the semantics of the input (Church & Hanks, 1989; DeRose, 1988; Garside *et al.*, 1987).

The input of a POS tagger is a string of words and a specified POS tagset. The output is a single best tag for each word. Typically, tags are also applied to punctuation marks. Each POS tag is composed of the part of speech or primary grammatical category of the word, and usually adds secondary grammaticaly information (number, gender, person, etc.) (Primary and secondary grammatical categories were introduced in Section 3.2). Typically, the set of POS tags is predefined by an expert human for a specific language. A well-known tagset is the Penn TreeBank tagset (Marcus *et al.*, 1993) (Table 3.1, page 40).

Assigning POS tags to unannotated text is not trivial (Santorini, 1990). Some words are ambiguous, in the sense that they have more than one grammatical role. For example, `book` can be a noun or a verb, as mentioned in Section 3.2. This is not rare: words of two or more possible POS tags account for over 10% of all the words in the Brown corpus (DeRose, 1988).

The difficulty in assigning the correct POS tag to a word can be overcome by looking at the context of the word. E.g. `book` in isolation is ambiguous, but in the context `to read a book`, it is a noun. Different POS tagging approaches use context in different ways. Initially, most POS taggers assign POS tags to words on the basis of a pre-specified word-tag lexicon. These initial POS tags are then altered according to contextual evidence. *Rule-based* POS taggers usually apply pre-specified rules:

$$pos_i = pos_j \text{ if } P \tag{3.8}$$

where pos_i is the POS tag assigned to a word initially, and pos_j is the POS tag assigned to the word under the contextual conditions described in P. *Stochastic* POS taggers generally alter the initial POS tags using word-tag probabilities ($P(word|pos)$), extracted from a human annotated corpus. *Transition probabilities* are probabilities of a POS tag given the previous POS tag, and *emission* *probabilities* are probabilities of a word given a POS tag. The probability of a POS sequence given a word sentence is then the product of the transition and emission probabilities involved.

Next, three standard POS taggers (Mihalcea, 2003) are presented, which are used later in the thesis (Chapter 6).

3.4.1 Transformation based (Brill) tagger

The **Transformation Based tagger** (aka *Brill tagger*) is a POS tagger that implements an approach to natural language processing called *transformation-based error-driven learning* (TBL) (Brill, 1995). TBL consists of learning transformation rules automatically (Equation 3.8 is an example of a transformation rule). The Brill tagger learns automatically POS transformation rules from a pre-tagged corpus and uses them to POS tag sequences of words. The Brill tagger uses *morphological* and *contextual* rules: morphological rules take into account word morphology (prefixes, suffixes, capitalization, and so on); contextual rules take into account the words and POS tags occurring before and after a given word. The Brill tagger shares features of both rule-based and stochastic tagging approaches: like rule-based taggers, it assigns POS tags to words according to rules; like stochastic taggers, it learns rules automatically from a pre-tagged training corpus. The Brill tagger is popular due to its accuracy and public availability (Mihalcea, 2003).

3.4.2 Maximum entropy tagger

The **Maximum Entropy tagger** (*Mxpost*) (Ratnaparkhi, 1996) is a stochastic POS tagger, which implements the Maximum Entropy principle of Rosenfeld (1994): the basic idea is that better use of context improves tagging accuracy. Mxpost models the probability of a tagged sequence of words using the transition and emission probabilities presented above. When deriving $P(word|pos)$ from a corpus, if a word is sparse, the resulting probabilities risk of being unreliable, so Mxpost uses morphological information for such cases: prefixes, suffixes, numbers, upper-case characters, or special symbols. The inference of Mxpost consists of the estimation of the parameters that combine these features and minimise the uncertainty in assigning a POS tag to a word. Mxpost has been popular because of its accuracy and rich set of contextual features (Mihalcea, 2003).

3.4.3 TreeTagger

The ***TreeTagger*** (Schmid, 1994) is a stochastic POS tagger, which uses transition and emission probabilities to model the probability of a tagged sequence of words, similarly to Mxpost. Rare or uncommon words can cause inaccurate estimations as mentioned above. Mxpost addresses this by using morphological information. The TreeTagger addresses this by estimating $P(word|pos)$ probabilities with binary decision trees (Schmid, 1994). Decision trees are trained on pre-tagged corpora, and output a set of questions that can be asked about a word to determine its correct POS tag. Decision trees are then built by finding the question whose resulting partition is the 'purest', splitting the training data according to that question, and then recursively reapplying this procedure on each resulting subset. The decision trees used in the TreeTagger are built recursively using a modified version of the ID3-algorithm (Quinlan, 1983). The TreeTagger is popular because of its accuracy and availability in languages other than English (Mihalcea, 2003).

3.5 Information retrieval applications using parts of speech

Parts of speech have been applied to many language processing applications. Most of these applications process first degree parts of speech differently to second and third degree parts of speech on empirical grounds, with the goal to maximise the performance of the process involved. In IR, these efforts were initiated in the 1980s, and intensified in the 1990s, reporting retrieval benefits. After that time, these efforts decreased: baseline system performance improved, and the cost associated with linguistic processing was not worth the small benefits over the already improved baselines (Tait, 2005).

Generally, part of speech information has been used for stemming, generating stopword lists, and identifying pertinent terms or phrases in documents and/or in queries. Some of these numerous applications of parts of speech in IR are listed below in chronological order. More information is found in the overviews by Karlgren (1993); Smeaton (1986, 1999); Tait (2005). Unless otherwise stated, most of the following applications report the use of parts of speech in IR systems that use the vector space model to match documents to queries. (The vector space model for retrieval was presented in Section 2.4.2.2.)

Dillon & Gray (1983) used parts of speech to identify and index phrases. They applied a set of rules which searched for part of speech patterns known to indicate content-bearing terms. They reported improvement in the performance of their FASIT system. Along the same lines, Salton (1988); Sparck Jones & Tait (1984) used parts of speech to select indexing terms, and reported similar results. Similarly to these studies, Fagan (1987) used parts of speech to identify and index 2-word and 3-word phrases. In addition, he also used statistical co-occurrence to identify and index phrases. He compared retrieval performance when using statistical versus syntactic phrases (identified by their part of speech). He found not much difference between the two in terms of retrieval effectiveness, though the statistical approach was computationally more economical.

Instead of processing parts of speech in documents in order to identify phrases for indexing, which is computationally expensive, Smeaton & van Rijsbergen (1988) used parts of speech to identify and index noun groups in queries, in order to get a better representation and understanding of the user information need. A syntactic parse of the query was used to identify dependent word pairs, and the retrieval strategy was to search for co-occurrence of word pairs within a sentence in documents. They reported improved retrieval performance.

In the 1990s, studies using parts of speech to identify indexing phrases became more elaborate: Lewis & Croft (1990) proposed to link syntactic and semantic information as follows: they used parts of speech to identify and index phrases, in which they also identified topical clusters. They reported small improvements in retrieval effectiveness. Similarly, Jacobs & Rau (1993) proposed using parts of speech to identify syntactic frames, i.e., relations between events and actions. They reported promising results, and their system was the highest-scoring system at the Message Understanding Conference (MUC) evaluation for 1993.

Apart from the above more elaborate efforts to combine parts of speech with semantic evidence, several studies using parts of speech for IR remained focussed on improving the selection of indexing keywords and phrases: Lin (1995) used parts of speech to identify and index compound nouns and adjective - noun collocations. He evaluated this approach as part of MUC 1995, and reported promising results. Evans & Zhai (1996) proposed a hybrid technique that identified and indexed noun phrases, by combining both part of speech and statistical co-occurrence. They reported improvement in retrieval recall and precision. This work was followed up by Arampatzis *et al.* (1997); Jacquemin *et al.* (1997); Zhai *et al.* (1997), who focussed on the use of nouns and adjectives for indexing, and reported improvement in retrieval performance.

A different approach to using parts of speech for IR was put forward by Zhai (1997), who proposed enhancing the IR system's index representation as follows: he added noun phrases to all terms in the index, in order to represent better what the document is about, and thus to improve retrieval performance. He reported that adding noun phrases to the index of all terms improved precision and recall of their IR system indeed.

Another approach, also using parts of speech with the IR system's index, was presented by Pederson *et al.* (1997), who indexed selectively based on parts of speech. They indexed nouns, verbs, adjectives, adverbs, interjections, numerals, abbreviations, and participles, and left out conjunctions, determiners, infinite markers, prepositions, and pronouns. In addition, they implemented a term weighting formula based on parts of speech, which favoured noun phrases and adjective phrases when computing term weights. They reported improvement in precision and recall over standard indexing of all terms.

In the same year, Strzalkowski & Lin (1997) used parts of speech to select indexing keywords, to remove stopwords from the index, and also to index phrases. Specifically, they removed all but nouns, verbs, adjectives, participles, adverbs, as well as some very frequent words. Similarly to Pederson *et al.* (1997), they used different weighting formulae for the various parts of speech, which they tuned to optimise retrieval performance. They also applied a term weighting formula that 'boosted' the term weights of words inside noun phrases. They reported an improvement in retrieval performance, but high computational costs.

The results of Strzalkowski & Lin (1997) were in line with another study of the previous year by Strzalkowski & Sparck Jones (1996), which concluded that *NLP has solid but limited impact on retrieval quality*. This observation was partially shared by Crestani *et al.* (1997), who showed that improvement in retrieval performance did not apply when indexing noun phrases alone and when using short queries (approximately three words long).

Generally, the overview of the Text Retrieval Evaluation Conference (TREC) for 1997 indicates that the state of the art in IR systems at the time used parts of speech to index or retrieve information (Voorhees & Harman, 1998).

Efforts to use parts of speech in selecting indexing keywords were also reported by Chowdhury & McCabe (1998), who compared indexing only nouns to indexing all parts of speech, and also to indexing everything but nouns. They used the following part of speech categories only: nouns, verbs[1], adjectives, adverbs, other, and constructed three indices:

[1] In this study, the category of verbs included participles.

- Indexing all terms (baseline).
- Indexing nouns only.
- Indexing only verbs, adjectives, adverbs, and other parts of speech, except nouns.

The index that included all parts of speech gave the best retrieval performance. The only-nouns index was a close second, and the everything-but-nouns index was markedly poor. Indexing only nouns overall reduced the system's performance by less than 1%, and gave system storage savings of around 9.5% for the index data.

In the same year, Flank (1998) proposed a different use of parts of speech for IR, namely, not only for selecting indexing keywords, but also when matching documents to queries. Her system, called Intermezzo, was deployed in a pre-product form at a government site and used a Boolean retrieval model. Parts of speech were used in two ways. First, part of speech patterns were used to identify multiword expressions (i.e., noun phrases). The identified patterns were then weighted differently than individual words. Second, incoming queries were POS tagged: only words that matched by part of speech the query terms were considered for matching by the system; if two or more parts of speech were possible for a particular word, the word was tagged with both. Also, the system implemented semantic expansion (using WordNet), which was constrained by part of speech information: terms were expanded with their synonyms from WordNet; however, only those expansions that applied to the correct part of speech in context were retrieved. The system also used parts of speech (combined with databases) for name recognition. Overall, results indicated that the combination of noun phrase syntax and name recognition improved recall by 18%, and that name recognition played a larger role in the improvement of the system performance than did noun phrase syntax.

In late 1990s, the general consensus seemed to be that simple statistical approaches were generally more effective than well-executed linguistically-motivated techniques for IR, such as elaborated use of parts of speech and semantics (Sparck Jones, 1999).

A different use of parts of speech was proposed by Narita & Ogawa (2000), who suggested using parts of speech for query construction. They examined the utility of phrases, extracted from query texts using parts of speech, as search terms for IR. They used single terms and phrasal terms in their query construction. They matched phrases of two terms using proximity constraints. They also associated lesser weight to phrasal terms than single terms, reasoning that the occurrence of a phrase in a document also indicates the occurrence of its constituent words. They reported some improvement in retrieval performance.

Studies using parts of speech to identify indexing keywords and phrases were also reported at that time (Fujita, 2001; Lin, 2001), which concluded that there were some retrieval benefits in the efficiency of the IR system, namely for rapid large-scale indexing.

More recently, Srikanth & Srihari (2003) used of parts of speech to identify concepts for IR. Using the language modelling approach (presented in Section 2.4.2.4), they proposed a *Concept Language Model*, which views the query as a sequence of concepts, and a concept as a sequence of terms. They assumed that concepts are phrases that are identified by the parts of speech of the query terms. They showed improved retrieval performance for some but not all queries. This approach differs from most previous techniques because it uses parts of speech solely to enhance the retrieval process of IR systems, and not the representation of document terms in the system index.

Parts of speech are also used consistently in other language processing applications, such as **summarisation** (see Mani (2001) for an overview), **semantic taxonomies** (Caraballo & Charniak, 1999), **question-answering** (Anick & Tipirneni, 1999), **text categorisation** (Jacobs, 1992). Parts of speech have also been used in **language teaching**, where nouns have been used for detecting language difficulty (Mikk, 2000, 2001), or in more general studies as the primary units for detecting textual content (Savicky & Hlavacova, 2002; Zubov, 2004).

3.6 Summary

This chapter introduced parts of speech as primary (or shallow) grammatical categories (Section 3.2), and presented a ranking of parts of speech (Section 3.3), which is used extensively in the rest of the thesis. Automatic ways for assigning parts of speech to words were also presented (Section 3.4), and in particular three standard POS taggers, which are used later in the thesis (Chapter 6), were discussed. Last, IR applications using parts of speech to enhance retrieval performance were overviewed (Section 3.5).

Chapter 4 introduces n-grams, and specifically POS n-grams.

Chapter 4

Part of speech n-grams

4.1 Introduction

This chapter presents part of speech (POS) n-grams. The chapter is organised as follows: Section 4.2 presents basic concepts and applications of n-grams. Section 4.3 presents basic concepts and applications of POS n-grams. Section 4.4 summarises and concludes this chapter.

4.2 n-grams

4.2.1 Basic concepts of n-grams

Given a contiguous sequence of items, an *n-gram* is a contiguous subsequence of that sequence. (This thesis follows the n-gram notation of Brown *et al.* (1992), where the subscript is the first item in the n-gram, and the superscript is the last item in the n-gram.) Let i_h^k be a contiguous sequence of items, where i_h is the first item, and i_k is the last item in the sequence. Then, i_j^{j+n-1} is a contiguous subsequence or string of that sequence, if $i_j \geq h$ and $n < k$. Such subsequences are called n-grams, usually when the number of items n in the string is fixed, and when they are extracted in a recurrent and overlapping way from the initial sequence (Damerau, 1971). For example, the 3-grams extracted from the initial sequence i_1^5 are: i_1^3, i_2^4, and i_3^5. The total of all sequences from which n-grams are extracted is often called *sample*, denoted S. n-grams are usually extracted from very large samples. The likelihood of observing an n-gram in the sample is assigned a probability, so that the more frequent the n-gram is in the sample, the higher its probability of occurrence. The computational mechanism for obtaining these probabilities is referred to as a *language model*[1] (Brown *et al.*, 1992). Hence, a language model is a probability distribution over sets of n-grams.

To compute the probability of occurrence of an n-gram in a sample, in the simplest case, its *relative frequency*[2] can be used. Let $c(i_j^{j+n-1})$ be the number of times that the n-gram i_j^{j+n-1} occurs in the sample S. Let $|S|$ be the number of all n-grams in the sample. Then, the relative frequency of i_j^{j+n-1} in the sample is:

$$RF_{i_j^{j+n-1}} = \frac{c(i_j^{j+n-1})}{|S|} \tag{4.1}$$

n-gram language models typically assume that the likelihood of an item depends only on its immediate context, i.e., the $n-1$ item before it (*Markov independence assumption* (Markov, 1913)). This assumption implies that the probability for an n-gram can be decomposed into the probabilities of smaller n-grams appearing in it:

$$P(i_j^{j+n-1}) = P(i_j) \cdot P(i_j|i_j^{j+1}) \cdot \ldots \cdot P(i_n|i_j^{n-1}) \quad (n > 2) \tag{4.2}$$

The value of n is called the *order* of the language model, and controls the amount of context captured inside the n-gram. As n increases, the accuracy of an n-gram model increases, but the reliability of the estimation decreases (because the number of n-grams in the sample decreases). Typically, the number of all possible n-grams in the sample is called the *complexity* of the language model. The complexity of a language model increases exponentially with n: if the sample contains $|i|$ items, then, for a fixed n, the number of all possible n-grams in the sample is:

$$\text{language model complexity: } |i|^n \tag{4.3}$$

In evaluating language models (stand-alone, not as part of another process), an intrinsic measure of their quality is their *perplexity*: given a sample S, and a number $|S|$ of possible n-grams in it, the perplexity of a language model is the number of probability predictions $P(S)$ that can be computed in total. (Equation 4.2 is an example of such probability predictions.) Mathematically, language model perplexity is the reciprocal of the geometric average of the probabilities of

[1] The language modelling approach to IR (Section 2.4.2.4, page 27) is related but not identical to this general model. The difference between the two is discussed in Section 4.2.2, page 52.

[2] Often, the terms *relative frequency* and *maximum likelihood* (Jurafsky & Martin, 2000) are used interchangeably (Manning & Schutze, 1999), under a relaxed definition of the latter (Aldrich, 1997).

the predictions in S:

$$\text{language model perplexity: } P(S)^{-\frac{1}{|S|}} \tag{4.4}$$

The smaller the perplexity, the more accurately the language model captures language regularities of the sample. Because perplexity depends not only on the language model but also on the sample, it is important that the sample be large and representative.

4.2.2 Applications of n-grams

An overview of applications using n-grams is presented. Applications using POS n-grams are presented separately later, in Section 4.3.3.

Early language models can be traced to Markov (1913), who used 2-grams and 3-grams of letters (*Markov chains*) to predict the occurrence of vowels and consonants in Russian. Shannon (1948) also used sequences of letters and words to measure the degree to which the English language can be compressed, with implications to coding and information theory. Feller (1950) and Gallager (1968) introduce n-grams for information theory.

Since the early efforts of Markov (1913) and Shannon (1948), n-grams have been used in many language processing applications. In **speech recognition** (Bahl & Jelinek, 1975; Bahl *et al.*, 1990; Baker, 1975; Jelinek, 1977; Rabiner, 1989), n-grams of phonemes have been used to predict the most likely phoneme combinations, hence the most likely sound. The aim in these approaches is to use part of speech information in order to identify word triggers or term dependence, that might be of help in predicting a spoken word. In general **parsing** (Suen, 1979) and **part of speech tagging** (Brown *et al.*, 1992; Cutting *et al.*, 1992; Ratnaparkhi, 1996; Schmid, 1997), n-grams of words and parts of speech have been used to predict the most probable part of speech, given its preceeding part of speech, and the most probable word, given its part of speech. In **language generation** (Ratnaparkhi, 2000), n-grams of letters or words have been used to create text automatically (for example, the dissociated press algorithm (Beeler *et al.*, 1972)). In **named entity identification** (Burger *et al.*, 1998), n-grams of words have been used to predict named entities (for example names of companies or persons). In **machine translation** (Mariòo *et al.*, 2006; Och & Ney, 2004), word n-grams have been used to model correspondences of words between languages, and hence to predict the most likely translation. In **language recognition**, for example the TextCat language guesser by Canvar

& Trenkle (1994), letter n-grams from different languages have been used as a feature to identify language. In **pattern recognition** (Hull & Srihari, 1982; Kim & Shawe-Taylor, 1994; McElwain & Evens, 1962), word n-grams have been used to predict word patterns. In **spelling correction** (Angell *et al.*, 1983; Ullmann, 1977; Zamora *et al.*, 1981), letter n-grams have been used to find candidates for the correct spelling of misspelled words. In **keyword indexing** for IR (Adams, 1991; Burnett *et al.*, 1979; Canvar, 1993, 1994; Cohen, 1995; Crowder & Nicholas, 1995; Feng *et al.*, 2000; Huffman & Damashek, 1994; Mehmet, 1990; Schuegraf & Heaps, 1973; Willett, 1979), word n-grams have been used to select indexing keywords (the indexing process was presented in Section 2.3.2). In **text compression** (Wiskiewski, 1987), word n-grams have been used to summarise text. In **clustering** (Collier, 1994a,b), word n-grams have used to identify the boundaries and order of different topics in text. In **text classification**, part of speech n-grams (Argamon *et al.*, 1998a,b; Johannes, 1998; Santini, 2007) have been used as a genre-revealing feature to predict the genre or domain of text.

Apart from identifying indexing keywords for IR, n-grams have also been used in IR to match documents with respect to a query, according to the likelihood that the query has been generated from the document (Croft & Lafferty, 2003; Hiemstra, 2001; Kraaij, 2004). Language models for IR were presented in Section 2.4.2.4. In short, the use of language models for IR is different to the above applications because it uses n-grams not to model language regularities, but simply as a mechanism to estimate the probability of a term given a document.

n-gram applications are found not only in academia, but also in industry: Google[1] announced using n-grams for machine translation, speech recognition, spelling checking, entity detection, and data mining, and, in 2006 released through the Linguistic Data Consortium a list of word n-grams compiled from in-house data.

n-grams have also been used in non-linguistic applications. In **genetic sequence analysis** (Cheng *et al.*, 2005; White *et al.*, 1993) n-grams of DNA sequences are used in genetic sequence search and to identify which species short DNA sequences were taken from (e.g., Basic Local Alignment Search Tool (BLAST) family of programs (Zhang & Madden, 1997)). In **image recognition** (Soffer, 1997), n-grams are used to extract features for clustering large sets of satellite earth images and for determining what part of the earth a particular image came from. In **machine learning** (Kurai *et al.*, 2006), n-grams are

[1] http://googleresearch.blogspot.com/2006/08/all-our-n-gram-are-belong-to-you.html

used to design kernels that allow machine learning algorithms (e.g., support vector machines) to 'learn' from string data. In **compression algorithms** (Teng & Neuhoff, 1995), n-grams are used to improve compression where a small area of data requires n-grams of greater length. In **optical character recognition** (OCR) and **intelligent character recognition** (ICR) systems (Harding *et al.*, 1997), character n-grams are used to identify correct sequences of characters.

To recapitulate, n-grams are contiguous subsequences of a contiguous sequence of items. Typically, n-grams are used to predict the occurrence of an item in a sequence (e.g., POS tagging), and/or to characterise the sample from which they were extracted (e.g., text classification).

4.3 Part of speech n-grams

4.3.1 Introduction

Section 4.2 introduced n-grams as contiguous subsequences of a sequence of items. The applications of n-grams presented showed that n-grams tend to be used to predict the occurrence of an item in a sequence, or to model the sample from which they were extracted. In this section, n-grams of contiguous part of speech (POS) sequences are presented, as well as the notation to be used in the rest of the thesis.

4.3.2 Definitions and notation

Let S be a sample, containing contiguous sequences of terms t (e.g., sentences). Then, for some fixed n, t_j^{j+n-1} is a term n-gram, where t_j is the first term, and t_{j+n-1} is the last term in the n-gram. For example, for S = `the cat sat on the mat`, and for n=3:

$$\begin{aligned} t_1^3 &= \texttt{the cat sat} \\ t_2^4 &= \texttt{cat sat on} \\ t_3^5 &= \texttt{sat on the} \\ t_4^6 &= \texttt{on the mat} \end{aligned}$$

Let pos be a part of speech, and ϕ be a function that maps term t_i to its part of speech pos_i, so that:

$$\phi(t_i) = pos_i \tag{4.5}$$

This can be done by any POS tagger (see Section 3.4)[1]. The relation between t_i and pos_i in Equation 4.5 is not symmetrical: knowing pos_i does not imply t_i. E.g., the term `cat` is a `noun`; but a `noun` corresponds to many different terms.

For some fixed n, pos_j^{j+n-1} is a POS n-gram, where pos_j is the first part of speech in the n-gram, and pos_{j+n-1} is the last part of speech in the n-gram. For each term n-gram, there exists a POS n-gram. This means that the order of the parts of speech inside a POS n-gram reflects the order of the words in the term n-gram. It follows that:

$$\phi'(t_j^{j+n-1}) = pos_j^{j+n-1} \tag{4.6}$$

where ϕ' denotes applying function 4.5 to every term inside t_j^{j+n-1}. Applying function 4.6 to the above term n-grams gives the following POS n-grams:

$$\begin{aligned}\phi'(t_1^3) &= pos_1^3\\ \phi'(t_2^4) &= pos_2^4\\ \phi'(t_3^5) &= pos_3^5\\ \phi'(t_4^6) &= pos_4^6\end{aligned}$$

For this example, the correspondence between POS n-grams and term n-grams is:

$$\begin{aligned}\phi'(\texttt{the cat sat}) &= \texttt{DT NN VB}\\ \phi'(\texttt{cat sat on}) &= \texttt{NN VB IN}\\ \phi'(\texttt{sat on the}) &= \texttt{VB IN DT}\\ \phi'(\texttt{on the mat}) &= \texttt{IN DT NN}\end{aligned}$$

The part of speech abbreviations used in this thesis are explained in Table 3.1, page 40.

Let $\{t_j^{j+n-1}\}_i$ be the set of all term n-grams t_j^{j+n-1} that contain term t_i *at any position inside the n-gram*. For example, given the above 3-grams:

$$\text{for } t_i = \texttt{cat},$$
$$\{t_j^{j+n-1}\}_i {=} \{\texttt{the cat sat, cat sat on}\}_{\texttt{cat}}$$

$$\text{for } t_i = \texttt{the},$$
$$\{t_j^{j+n-1}\}_i {=} \{\texttt{the cat sat, sat on the, on the mat}\}_{\texttt{the}}$$

[1] Parts of speech are assigned to terms in the context of sentences or phrases, not individually as shown above. This point is discussed in Section 3.4.

Then,

$$\phi''(\{t_j^{j+n-1}\}_i) = \{pos_j^{j+n-1}\}_i \tag{4.7}$$

where ϕ'' denotes applying function 4.6 to each term n-gram in the set $\{t_j^{j+n-1}\}_i$. For example,

$$\phi''(\{\texttt{the cat sat, cat sat on}\}_{\texttt{cat}}) = \\ \{\texttt{DT NN VB, NN VB IN}\}_{\texttt{cat}}$$

$$\phi''(\{\texttt{the cat sat, sat on the, on the mat}\}_{\texttt{the}}) = \\ \{\texttt{DT NN VB, VB IN DT, IN DT NN}\}_{\texttt{the}}$$

The relation described by function 4.7 is important for computing a term weight from POS n-grams, because it maps a term to all the POS n-grams that 'contain'[1] it. How this is done is shown in Section 5.3.

The next section presents applications of POS n-grams.

4.3.3 Applications of part of speech n-grams

Section 4.2.2 presented an overview of applications of n-grams, including POS n-grams. Here, applications of POS n-grams are discussed in detail.

A common application that uses POS n-grams is stylometric **text categorisation**, which consists of making predictions about the author or genre of a given text (Lim *et al.*, 2005). (See Sebastiani (2002) for an overview of approaches and problems of text categorisation.)

Baayen *et al.* (1996) used POS n-grams, which they described as 'pseudo-word sequences', to measure syntactic differences among texts. They then used these differences for automatic authorship attribution. Their motivation was that particular document classes can favour certain syntactic structures. They used automatic parsing to classify input texts. They reported promising results and concluded that part of speech information is at least as revealing as any other stylistic text classification feature, for example lexical information.

Argamon *et al.* (1998a,b) suggested the use of POS 3-grams, which they called *part of speech triplets*, for text classification, as a shallow approach to syntax. Their approach was presented as a computationally inexpensive and robust alternative to the idea of Baayen *et al.* (1996) of using syntactic structures for text classification. Their rationale was that POS 3-grams are large enough to encode

[1] POS n-grams 'containing' a term = POS n-grams corresponding to term n-grams containing a term.

useful information, yet small enough to be computationally manageable. They used POS 3-grams and function words for automatic text classification. POS tagging was done with the Brill tagger (presented in Section 3.4.1), and POS 3-grams included punctuation. For classification they used only POS 3-grams that occurred in between 25% and 75% of all the documents; the rest POS 3-grams were ignored. They concluded that the use of POS 3-grams for text classification was promising, but not entirely conclusive, hence that further experimentation and variation of experimental settings was needed.

Studies by Johannes (1998) showed similar conclusions. Efforts were also made to use POS n-grams for author gender classification (Koppel *et al.*, 2003a,b). In these studies, POS n-grams were used together with function words and machine learning techniques to identify the gender of authored text. POS n-grams were described as ‘quasi-syntactic features’, and were shown to be promising classification features.

Santini (2007) extended the work of Argamon *et al.* (1998a,b), by using POS n-grams for text classification. She used POS n-grams on their own, not in combination with other features (e.g., function words) as reported in Argamon *et al.* (1998a,b). Additionally, she varied n between 1-3, and experimented with the classification of both spoken and written genres included in the British National Corpus (BNC) (Aston & Burnard, 1998). She used POS n-grams, both including and excluding punctuation, and she selected POS n-grams with a frequency of occurrence between 30-100 in a single genre collection. She showed that POS n-grams excluding punctuation resulted in higher classification accuracy for the written genres, but that including punctuation improved the classification accuracy for spoken genres. She also confirmed the reasoning of Argamon *et al.* (1998a,b) about the order $n = 3$ of POS n-grams, by showing that, overall, POS 3-grams resulted in better classification accuracy than POS 2-grams and POS 1-grams. Overall, she showed that POS n-grams have strong discriminating power as features for text classification.

Even though the above studies have reported promising results in using POS n-grams for text classification, there also exist studies claiming that the discriminating power of POS n-grams for text classification is limited and prone to noise (Aaronson, 1999; Kessler *et al.*, 1997). These studies refer to the unrestricted use of POS n-grams for classification, i.e., without selecting POS n-grams of certain frequencies, or removing punctuation, as reported in Argamon *et al.* (1998a,b); Santini (2007) and most other studies using POS n-grams for classification. The conclusion from the above seems to be that POS n-grams, when selected carefully, can be stable indicators of style and hence useful for text classification.

4.4 Summary

This chapter introduced n-grams (Section 4.2), and particularly POS n-grams (Section 4.3). Basic definitions, notation, and main assumptions were introduced. Also, applications using n-grams, and POS n-grams in particular, were overviewed.

Chapter 5 discusses the relation between POS n-grams and informative content, which is central in this thesis.

Chapter 5

Part of speech n-grams and informative content

5.1 Introduction

This chapter discusses the link between POS n-grams and informative content, which is the main motivation of this work. Firstly, the relationship between POS n-gram frequency and informative content is presented. Secondly, based on this relationship, a methodology is presented for using POS n-grams to compute how informative a word is in general, not with respect to a topic. This is called Part of Speech Information Score (PIS) and this chapter develops two alternative ways of calculating this (denoted by PIS_1 and PIS_2).

This chapter is organised as follows. Section 5.2 presents the relation between POS n-gram frequency and informative content. Section 5.3 presents the general methodology for computing PIS, and discusses the reasoning behind this computation. Section 5.4 summarises and concludes this chapter.

5.2 Frequency and informative content of part of speech n-grams

The initial motivation for using POS n-grams in IR was the empirical observation that, when ranking all the POS n-grams in a collection according to their frequency, the most frequent POS n-grams have a tendency to contain mostly open class parts of speech, and the least frequent POS n-grams have a tendency to contain mostly closed class part of speech. (Open and closed parts of speech were presented in Section 3.2.)

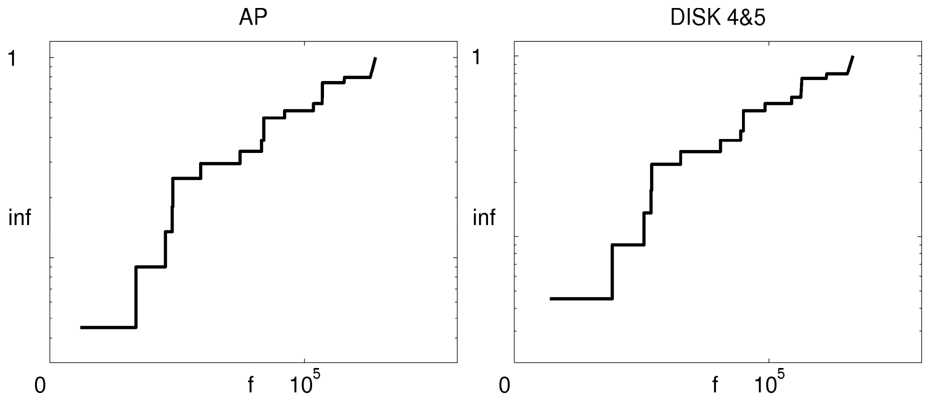


Figure 5.1: Frequency versus informative content of POS 4-grams (AP, Disks 4&5).

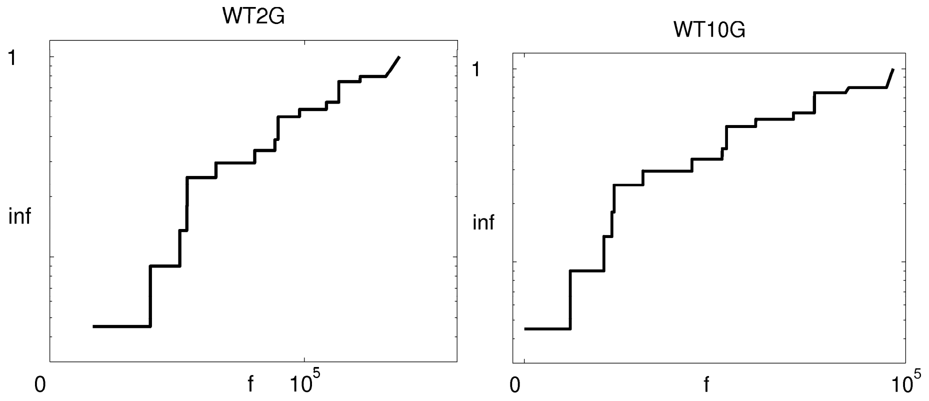


Figure 5.2: Frequency versus informative content of POS 4-grams (WT2G, WT10G).

This observation made sense: there exists a much greater number of open class than closed class words in language (Francis & Kuĉera, 1982; Tuldava, 1996), i.e., compare the number of nouns to the number of determiners (Hudson, 1994). Also, by definition, open class words tend to convey meaning, hence they are necessary in language, whereas closed class words tend to modify existing meaning, hence they can become redundant (Miller, 1951). A common illustration of this aspect of language is the predominant use of nouns in user queries on the Web (Ozmutlu *et al.*, 2004).

To illustrate this observation, the frequency of POS n-grams in a collection (f) is plotted against how informative POS n-grams are (inf) (Figures 5.1 - 5.3). The informative content of POS n-grams is measured by counting how many informative parts of speech are contained in them (see Algorithm 3). In

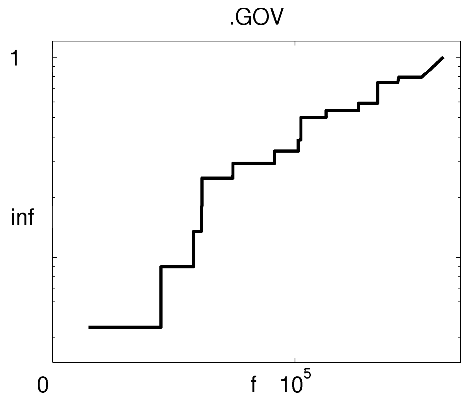


Figure 5.3: Frequency versus informative content of POS 4-grams (.GOV).

this context, 'informative content' refers to gaining awareness of content, as opposed to identifying specific semantic properties (e.g., topic). Hence, this type of informative content is by definition non-topical.

Algorithm 3 Counter of informative content for POS n-grams

```
for each POS $n$-gram do
    allocate a counter $inf$
    set $inf \leftarrow 0$
    for each part of speech $pos$ in the POS $n$-gram do
        if $pos \in$ open class then
            set $inf \leftarrow inf + 1$
        end if
    end for
end for
```

Figures 5.1 - 5.3 plot POS n-gram frequency (x axis) against informative content (y axis) for POS 4-grams extracted in five different standard TREC collections, namely AP, Disks 4&5, WT2G, WT10G, .GOV. These collections are used extensively in the experiments reported in Chapters 6-7, and are presented in details in Table 6.3, page 80. Figures 5.1 - 5.3 show that when f increases, inf increases too, and vice versa. This observation is also valid for POS 5-grams, the corresponding figures of which are presented in Appendix C, Figures C.1 - C.3, pages 167 - 167.

Algorithm 3 is a simple heuristical way of looking at how informative POS n-grams are; other heuristics, which for instance distinguish between different open and closed class parts of speech and penalise the presence of closed class parts

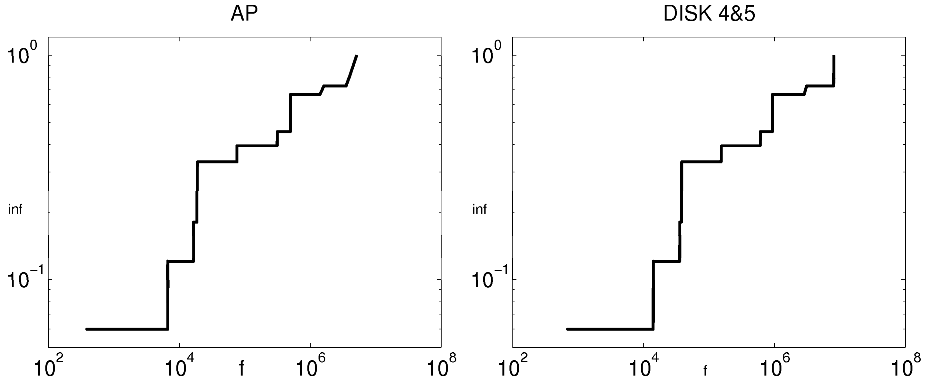


Figure 5.4: Frequency versus informative content of POS 4-grams (AP, Disks 4&5).

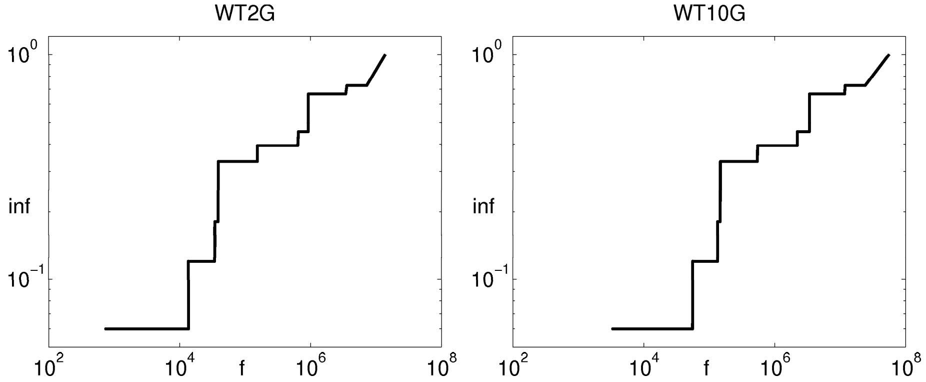


Figure 5.5: Frequency versus informative content of POS 4-grams (WT2G, WT10G).

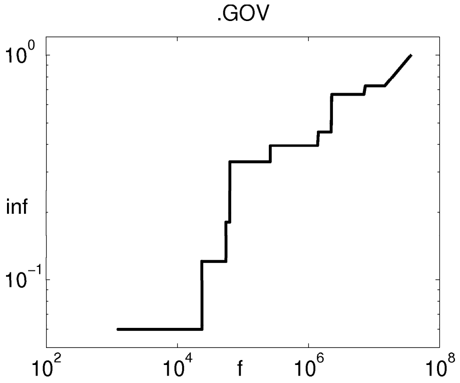


Figure 5.6: Frequency versus informative content of POS 4-grams (.GOV).

of speech inside the POS n-gram (such as Algorithm 4 for instance), have also produced very similar findings: Figures 5.4 - 5.6 plot POS 4-gram frequency (x axis) against informative content (y axis) for POS 4-grams, where the informative content of POS 4-grams has been computed with Algorithm 4. Similarly to Figures 5.1 - 5.3, Figures 5.4 - 5.6 also show that when f increases, inf increases too, and vice versa.

Algorithm 4 Refined counter of informative content for POS n-grams

```
for each POS $n$-gram do
    allocate a counter $inf$
    set $inf \leftarrow 0$
    for each part of speech $pos$ in the POS $n$-gram do
        if $pos \in$ open class then
            if $pos \in$ first degree then
                set $inf \leftarrow inf + 1$
            else if $pos \in$ rank then
                set $inf \leftarrow inf + \gamma$ $(0 < \gamma < 1)$
            end if
        end if
        if $pos \in$ closed class then
            set $inf \leftarrow inf - 1$
        end if
    end for
end for
```

The observation that POS n-gram frequency is rougly proportional to how informative a POS n-gram is, is the motivation for using POS n-grams in IR.

5.3 Part of speech information score for terms

This section introduces a framework for deriving a term information score (called PIS) exclusively from POS n-grams, based on one hand on the finding that POS n-gram frequency and informative content are approximately directly proportional, and on the other hand on the rankings of parts of speech presented in Sections 3.2-3.3. The main motivation of PIS is that the more often a term co-occurs with other informative terms, and the higher the rank of the part of speech of the term, the more informative that term is likely to be. Term co-occurrence information is derived from POS n-grams, and the frequency of this co-occurrence is derived from POS n-gram frequency. The type of informative content measured by PIS is non-topical, and it applies to individual terms, i.e., like a term weight.

To compute PIS, these two types of information are combined:

- Part of speech class and rank
- POS n-gram statistics

POS class and rank represent a priori information about non-topical content in language (how informative a part of speech is, in general); POS n-gram statistics represent observed information about language structure (how words[1] co-occur in general). To compute PIS, the above two types of information are combined using basic probabilities. See Good (1968) for an introduction into probabilistic reasoning, and Best (2001) for probability distributions in language.

5.3.1 General methodology

The general methodology for computing PIS is as follows:

- **Step 1.** Approximation of the probability that an individual part of speech is informative, using part of speech class and rank information. E.g., how informative an individual noun or verb is. (Section 5.3.2).
- **Step 2.** Extension of step 1 to approximate the probability that a POS n-gram is informative. E.g., how informative a POS n-gram is, on the basis of how informative its part of speech components are. (Section 5.3.3).
- **Step 3.** Extension of step 2 to approximate the probability that an individual term is informative, by mapping term n-grams containing this term to their corresponding POS n-grams:
  1. For a term, get all the term n-grams that contain it.
  2. Map all these term n-grams to their corresponding POS n-grams.
  3. The total probability that these POS n-grams are informative gives PIS. (Step 3 is described in Section 5.3.4).

The methodology shown above is one of several possible ways of computing PIS. Alternative computations of the probabilities shown above are also possible. For instance, the probabilities of informative content for individual parts

[1] Part of speech classes of words, in particular.

POS	**class**	**degree**	**relation**
NN	open $= \{pos_o\}$	first $= \{pos'\}$	$\{pos'\} \subset \{pos_o\}$, Eq. 3.4
JJ,VR,VB	open $= \{pos_o\}$	second $= \{pos''\}$	$\{pos''\} \subset \{pos_o\}$, Eq. 3.5
RB	closed $= \{pos_c\}$	third $= \{pos'''\}$	$\{pos'''\} \subset \{pos_c\}$, Eq. 3.6
rest	closed $= \{pos_c\}$	-	-

Table 5.1: Classes, degrees & relations of primary parts of speech.

of speech are derived above from Jespersen's Rank Theory, a linguistic theory, and as such they are to a large extent empirical approximations. Alternatively, more mathematically accurate probabilities could be derived using statistics instead of Rank Theory. Also, the probability of informative content of a POS n-gram shown above is computed by decomposing it to the probabilities of its individual components, whereas it could also be derived by considering the n-gram as a whole. This and other alternative computations of PIS are discussed in Section 8.2.3.

The next section presents how the probabilities involved in computing PIS are derived.

5.3.2 Probability that a part of speech is informative

This section presents how to approximate the probability that an individual part of speech is informative (Step 1 in the methodology for computing PIS).

Let inf be an event of informative content, and pos be an event of an individual POS. Then $P(inf|pos)$ is the conditional probability that inf occurs given pos, or more simply the probability of pos being informative ($0 \leq P(inf|pos) \leq 1$).

Recall the set theoretic notation of part of speech classes (Section 3.2) and ranks (Section 3.3), pages 38-41, also summarised in Table 5.1 for easy reference. Then, drawing from the principles of Rank Theory,

$$0 < P(inf|pos_o) \leq 1 \tag{5.1}$$

where pos_o is an open class part of speech, and

$$P(inf|pos_c) = 0 \tag{5.2}$$

where pos_c is a closed class part of speech. Equation 5.1 states that there is always some probability of an open class part of speech being informative (> 0).

Equation 5.2 states that there is no probability of a closed class parts of speech being informative (= 0). These are assumptions that do not always hold: for example, a closed class part of speech can be informative. These assumptions are in line with Rank Theory, and are made as approximations.

Open class parts of speech can be either first degree or second degree parts of speech (Equations 3.4, 3.5, Table 5.1, and Equation 3.7, page 42.) Equation 5.1 can be further modified to approximate the probability that an open class part of speech is informative, differently for first degree - open class parts of speech and second degree - open class parts of speech, according to Jespersen's Rank Theory[1].

Assuming that a first degree part of speech (pos') is always informative, and that a second degree part of speech (pos'') is always informative, but less than a first degree part of speech:

$$P(inf|pos') = \lambda \tag{5.3}$$

$$P(inf|pos'') = \varrho \tag{5.4}$$

$$(0 < \varrho < \lambda \leq 1)$$

Equation 5.3 states that there is always a probability that nouns (Jespersen's first degree) are informative. Equation 5.4 states that the probability of verbs, participles, and adjectives (Jespersen's second degree) being informative is shared by verbs, participles, and adjectives equally and can never be the maximum or minimum. These are assumptions which do not always hold, and which are made here as approximations.

Equations 5.3-5.4 define the probability of a first and second degree part of speech being informative as λ and ϱ, where $0 < \varrho < \lambda < 1$. Two ways of computing λ and ϱ are suggested:

- λ and ϱ can be tuned to optimise the performance of a process (presented in Section 5.3.2.1).
- λ and ϱ can be derived probabilistically (presented in Section 5.3.2.1).

In this thesis, both alternatives are implemented in Chapter 7.

[1] Alternatively, the assignment of probabilities to different POS classes could be realised according to POS statistics. Even though this thesis does not use statistics to compute the probabilities of POS classes, this point is discussed as a future extension in Section 8.2.3.

5.3.2.1 Tuning λ and ϱ to optimise system performance

The aim is to assign λ and ϱ values that reflect the probability that a first and second degree part of speech respectively is informative. Doing so will allow the computation of a term information score called PIS. If PIS is used as part of an overall process, the performance of which can be measured, λ and ϱ can be tuned to optimise this performance. For example, if PIS is used by an IR system, λ and ϱ can be tuned to maximise the Mean Average Precision (MAP) of the IR system. (MAP was presented in Section 2.7.)

$$\underset{\varrho|\lambda}{argmax} = MAP(\varrho|\lambda) \tag{5.5}$$

In Chapter 7, PIS is used as part of an IR system, and λ and ϱ are tuned to optimise MAP. The resulting λ and ϱ values are shown in Tables E.1 and E.2, Appendix E, pages 176 - 177.

5.3.2.2 Deriving λ and ϱ probabilistically

Another way to set λ and ϱ is to use Bayes rule and the probabilities of individual parts of speech being informative, defined in Equations 5.3-5.4.

$$P(inf|pos') = \lambda \Rightarrow \tag{5.6}$$

$$\frac{P(pos'|inf)P(inf)}{P(pos')} = \lambda \Rightarrow \tag{5.7}$$

$$P(pos'|inf)P(inf) = \lambda P(pos') \tag{5.8}$$

$$P(inf|pos'') = \frac{P(pos''|inf)P(inf)}{P(pos'')} \tag{5.9}$$

$$= \frac{[1 - P(pos'|inf)]\, P(inf)}{P(pos'')} \tag{5.10}$$

$$= \frac{P(inf) - \lambda P(pos')}{P(pos'')} \tag{5.11}$$

Without assuming any prior knowledge of $P(inf)$,

$$P(inf) = P(\overline{inf}) \Rightarrow P(inf) = 0.5 \tag{5.12}$$

Then, fixing a value for λ gives the value for ϱ.

This section approximated the probability that an individual part of speech is informative using Jespersen's Rank Theory. The next section approximates the probability that a POS n-gram is informative on the basis that its member parts of speech are informative.

5.3.3 Probability that a part of speech n-gram is informative

This section presents how to approximate the probability that a POS n-gram is informative (Step 2 in the methodology for computing PIS, Section 5.3.1). This probability is used later (Section 5.3.4) to compute how informative a term is (PIS).

How informative a POS n-gram is can be estimated on the basis of how informative its member parts of speech are. For each member of a POS n-gram, the probability of how informative it is can be computed with Equations 5.3-5.4, page 66. The combination of these probabilities gives an approximation of how informative the POS n-gram is.

Let pos_j^{j+n-1} be a POS n-gram (this notation was introduced in Section 4.3.2). Then, the probability that pos_j^{j+n-1} is informative $P(inf|pos_j^{j+n-1})$ can be approximated by averaging the probabilities of each of its members being informative:

$$P(inf|pos_j^{j+n-1}) \approx \frac{1}{n} \sum_{j=1}^{n-1} P(inf|pos_j) \tag{5.13}$$

where $P(inf|pos_j)$ is an approximation of the probability that an individual part of speech is informative, computed using Equations 5.3 & 5.4, page 66. Closed class parts of speech can be excluded from the computation of Equation 5.13, because they are assumed to be non-informative always (Equation 5.2, page 65). Note that this approximation does not apply to POS 1-grams.

In Equation 5.13, probabilities are combined linearly; there exist other alternatives to this, for instance computing their product or summing their logarithms. Generally, these alternatives are considered approximately equivalent. This point is discussed further in Section 8.2.4.

This section presented a way of approximating the probability that a POS n-gram is informative. The next section uses this probability to estimate how informative a term is.

5.3.4 Probability that a term is informative

This section presents how to approximate the probability that a term is informative, which is referred to as PIS (Step 3 in the methodology for computing PIS, Section 5.3.1).

For a term, PIS is estimated by doing two things:

1. All the term n-grams in which the term occurs are mapped to their corresponding POS n-grams.

2. The probabilities that each of these POS n-grams is informative are combined.

Two different computations of PIS are proposed, referred to as PIS$_1$ and PIS$_2$, in the rest of the thesis. PIS$_1$ is presented in Section 5.3.4.1, and PIS$_2$ is presented in Section 5.3.4.3.

5.3.4.1 Part of speech information score - PIS$_1$

The set of all POS n-grams which correspond to a term n-gram containing term i was defined in Section 4.3.2, Equation 4.7, page 56, as $\{pos_j^{j+n-1}\}_i$. Using this, the probability of a term t_i being informative can be approximated as:

$$P(inf|t_i) \approx \frac{1}{|C|} \sum_{j=1}^{n-1} P(inf|\{pos_j\}_i) \tag{5.14}$$

where

- $P(inf|\{pos_j\}_i)$ is computed with Equation 5.13 by replacing pos_j with $\{pos_j\}_i$, and

- $|C|$ is the number of all POS n-grams in the collection.

Equation 5.14 states that the probability of a term being informative is a function of how informative and how many are the POS contexts (POS n-grams) in which it occurs. The reasoning behind Equation 5.14 is that a term that occurs in many term n-grams, which themselves correspond to informative POS n-grams, is likely to be informative. This is quantified by combining the informative content of these POS n-grams, and their probability of occurrence.

More simply, PIS can be seen as the ratio of

$$\text{PIS} = \frac{\text{how informative all POS } n\text{-grams `containing' a term are}}{\text{how many POS } n\text{-grams occur in the collection}}[1] \tag{5.15}$$

In this thesis, the numerator is derived from basic principles of linguistics (presented in Sections 3.2 & 3.3), but alternatively it can also be derived from POS statistics (discussed in Section 8.2.3). The denominator is a frequency count of the total number of POS n-grams in the collection.

This section defined PIS$_1$ as the probability that a term is informative in general using only POS n-grams. The next section discusses PIS$_1$.

5.3.4.2 Discussion

The use of POS n-grams to compute a term information score, shown in Equation 5.14, begs the question: Why compute how informative POS n-grams are, and not simply how informative individual parts of speech are? This would mean using individual parts of speech only, and no POS n-grams at all. Equation 5.14 would then become:

$$\begin{align} P(inf|t_i) &\approx \frac{1}{|C|}\sum_{j=1}^{n-1} P(inf|\{pos_j\}_i) \tag{5.16}\\ &\approx \frac{P(inf|pos_i)}{|C|} \tag{5.17}\\ \tag{5.18}\end{align}$$

where

- $P(inf|pos_i)$ is the probability that the part of speech corresponding to term t_i is informative (computed with Equations 5.2, 5.3, and 5.4), and
- $|C|$ is the number of all terms in the collection.

Given that $0 \leq P(inf|pos_i) \leq 1$,

$$\begin{align} P(inf|t_i) &\approx \frac{P(inf|pos_i)}{|C|} \tag{5.19}\\ &\approx \log|C| \tag{5.20}\end{align}$$

[1] To be precise, POS n-grams 'containing' a term are POS n-grams that correspond to term n-grams which contain a term.

term	**f**$_{t,c}$	**PIS**$_1$	**term**	**f**$_{t,c}$	**PIS**$_1$
recall	2	0.3343	lyric	22,001	0.3134
tours	2	0.4388	tube	31,383	0.3496
linkin	56	0.5882	jose	35,629	0.4163
facebook	97	0.4114	dental	36,821	0.2886
mary	143	0.5005	symptom	36,881	0.3453
lady	164	0.3937	anderson	38,907	0.3884
you	214	0.2848	van	84,607	0.3701
paris	276	0.4885	girl	104,386	0.3306
boy	364	0.5016	aol	129,000	0.3292
mattel	672	0.4582	yahoo	138,589	0.3645
walmart	684	0.4312	hot	138,796	0.2402
jenna	757	0.4952	weather	155,278	0.3032
halen	1,201	0.5173	radio	156,908	0.3315
jameson	1,549	0.4261	station	162,711	0.3217
hotmail	1,684	0.4600	english	177,158	0.2645
sonia	1,743	0.4936	west	219,494	0.2686
bikini	1,818	0.4153	white	234,691	0.2509
play	2,484	0.2829	video	251,345	0.2913
nile	5,175	0.4136	park	284,315	0.2828
umbrella	5,450	0.3823	job	343,179	0.3138
nigeria	5,475	0.3982	game	359,590	0.3109
porn	5,744	0.4013	care	383,359	0.2300
depot	7,115	0.4143	music	411,467	0.2710
hilton	7,563	0.4365	local	467,758	0.2136
cheat	8,414	0.2598	free	523,669	0.2050
pamela	8,761	0.4470	find	524,453	0.1990
fever	11,894	0.3711	world	773,497	0.2891
gospel	14,946	0.3720	mail	855,685	0.2423
cnn	15,971	0.3895	name	901,525	0.2275
msn	16,788	0.3678	home	1,365,190	0.2567

Table 5.2: Example: terms, their frequency in WT10G, and their part of speech information score.

Hence, by using single parts of speech, instead of POS n-grams, the term information score would not model the 'part of speech context' in which terms occur, but it would only be a simple function of part of speech frequency in the collection.

On the contrary, by using POS n-grams instead of individual parts of speech to compute PIS$_1$, all the POS n-grams 'containing' a term in a collection are considered, and hence all the 'part of speech contexts' in which a term occurs are modelled. These 'part of speech contexts' contribute to PIS_1 the following: how informative the terms co-occurring with a given term are. The more informative these co-occurring terms, the higher the value of PIS_1. Also, the more often such terms co-occur, the higher the value of PIS_1. Hence, PIS_1 does not correspond to a 'flat' score for all terms of the same part of speech. Table 5.2 illustrates this point by showing the PIS_1 values of sample terms[1]. These term scores have been computed using Equation 5.14, page 69, with POS n-grams extracted from the WT10G TREC collection, which is presented later, in Table 6.3, page 80. These weights take into account all the 'part of speech contexts' in which these terms occur in WT10G. The term frequency in the collection is also presented for comparison with PIS_1.

Table 5.2 shows that overall, term frequency in the collection and PIS_1 tend to agree, however, they are not identical. For instance, several terms of similar frequency in the collection and/or of the same part of speech have different PIS_1. For example:

- `recall` - `tours`: same frequency (2), same part of speech (noun[2]), different PIS_1 (0.3343 - 0.4388);
- `mary` - `lady`: similar frequency (143 - 164), same part of speech (noun[3]), different PIS_1 (0.5005 - 0.3937);
- `jose` - `dental` - `symptom`: similar frequency (35,629 - 36,821 - 36,881), different part of speech (noun - adjective - noun), different PIS_1 (0.4163 - 0.2886 - 0.3453).

Conversely, there exist terms of similar PIS_1, but of different frequency and/or part of speech. For instance:

- `you` - `world`: similar PIS_1 (0.2848 - 0.2891), different frequency (214 - 773,497), different part of speech (pronoun - noun).

These examples illustrate the point that PIS_1 is more than a simple a function of the part of speech of a term and its term frequency in the collection. This does

[1] These terms were taken from the top 500 search engine keywords of the wordtracker Web site on 22/08/2007: http://www.searchengineguide.com/wt/2007/0822_wt1.html

[2] Either of these terms can also be a verb.

[3] To be precise, `mary` is a proper noun. This thesis does not distinguish between different noun classes, as described in Section 3.2, page 38.

not imply that the frequency of a term in the collection does not impact the computation of PIS$_1$, but rather that this impact is not a deciding factor. The next section presents an alternative computation of PIS, called PIS$_2$, for which the term frequency in the collection impacts the overall computation more.

5.3.4.3 Part of speech information score - PIS$_2$

Equation 5.14, page 69, computes the probability that a term is informative as a function of how informative are the POS n-grams containing the term and the total number of POS n-grams in the collection. Equation 5.14 can be simplified to produce a different estimation of how informative a term is ($P(inf|t_i)'$) as follows:

$$P(inf|t_i) \quad \approx \tfrac{1}{|C|} \textstyle\sum_{j=1}^{n-1} P(inf|\{pos_j\}_i)$$

can become

$$P(inf|t_i)' \quad \approx \tfrac{1}{|\{pos\}_i|} \textstyle\sum_{j=1}^{n-1} P(inf|\{pos_j\}_i) \tag{5.21}$$

where

- $P(inf|\{pos_j\}_i)$ is computed with Equation 5.13, page 68, by replacing pos_j with $\{pos_j\}_i$, (exactly as in PIS$_1$), and
- $\mid \{pos\}_i \mid$ is the total number of $\{pos\}_i$ parts of speech[1] in the collection (differently to PIS$_1$).

Equation 5.21 states that the probability that a term is informative corresponds to how informative are the POS n-grams containing the term and how many these POS n-grams are in the collection. This is called part of speech information score (PIS$_2$). Hence, this is the average probability that a POS n-gram containing the term is informative. More simply, PIS$_2$ is the ratio of

$$\text{PIS}_2 = \frac{\text{how informative all POS } n\text{-grams `containing' a term are}}{\text{how many POS } n\text{-grams `contain' a term}} \tag{5.22}$$

[1] Given a term t_i, $\{pos\}_i$ are its corresponding parts of speech, see Function 4.7, page 56.

As mentioned in Section 5.3.4.2, the numerator is derived from Jespersen's Rank Theory (Jespersen, 1913, 1929). The denominator is a frequency count, and specifically an 'inflated' estimation of the term frequency in the collection. It is inflated because, if a term occurs once in a sentence, it will occur repeatedly in n-grams extracted from that sentence[1].

5.3.4.4 Discussion

The computation of PIS_2 is reminiscent of the computation of inverse document frequency (IDF), presented in Section 2.3.2.2, Equation 2.1, page 16: On one hand, the IDF computation of a word looks at how many documents, in a general collection, contain the word. The intuition is that a word occurring in many documents is not likely to be very informative. On the other hand, the computation of PIS_2 looks at how many POS n-grams in a collection 'contain' a word. The intuition is that a word occurring in many *and* informative POS n-grams is likely to be informative. How informative POS n-grams are is computed by looking at their components. Hence, PIS_2 is expected to be more correlated to IDF than PIS_1, because the frequency of a term in the collection, which is central in IDF (namely in the denominator of Equation 2.1 as the number of documents that contain a term), also impacts the computation of PIS_2 (namely in the denominator of Equation 5.21).

Nevertheless, the differences between IDF and PIS_2 (and PIS_1 for that respect) are clear:

- IDF approximates the power of a term in discriminating between documents, whereas PIS_1 & PIS_2 approximate the non-topical informative content in a term, regardless how many documents the term occurs in.
- IDF uses lexical statistics (word/document counts), whereas PIS_1 & PIS_2 use shallow grammarical statistics.
- IDF is a bag-of-words measure (it does not consider term context), whereas PIS_1 & PIS_2 consider the 'part of speech context' of a term.

[1] This inflated estimation does not apply to terms at the start or end of a sentence, because these terms occur in one n-gram, unless they are repeated in the sentence.

5.4 Summary

This chapter presented the approximately proportional relationship between the frequency and informative content of POS n-grams (Section 5.2), and introduced a non-topical term score for approximating the probability that a term is informative using POS n-grams (Section 5.3). Two alternatives of this part of speech term information score were presented (PIS$_1$ and PIS$_2$), collectivelly referred to as PIS. The reasoning behind the computation of PIS as well as how it differs from an established term information score that uses lexical statistics (IDF) were also presented.

Next, Chapter 6 studies the statistical properties of POS n-grams and PIS in different collections, and with respect to IDF.

Chapter 6

Distribution of part of speech n-grams

6.1 Introduction

Chapter 5 discussed the relationship between POS n-gram frequency and informative content, and also proposed a term information score (PIS) that approximates the probability that a term is informative from POS n-grams. This chapter looks at the distribution of POS n-grams in different collections, and also compares PIS to IDF, by looking at their correlations in different collections.

Thic chapter is organised as follows. Section 6.2 presents the aims of this study. Section 6.3 presents the settings used, and shows that they are not biased. Section 6.4 presents the methodology followed. Section 6.5 presents and discusses the results of this study. Section 6.6 summarises and concludes this chapter.

6.2 Aims and anticipated outcomes

This study looks at POS n-gram distribution and compares the part of speech information score proposed in this thesis (PIS) to an established term information score (IDF). The aim is to see how POS n-grams are distributed in language, and also if PIS is correlated to IDF. Looking at the distribution of POS n-grams is motivated by the empirical finding that there seems to be a relationship between POS n-gram frequency and informative content. Finding a consistent POS n-gram distribution across collections would indicate that the relationship between POS n-gram frequency and informative content is not collection-dependent. Also, looking at a possible correlation between PIS and IDF can show the extent to which these two term scores differ from each other in modelling how informative a term is. Both PIS alternatives (PIS_1 and PIS_2) are compared to IDF.

6.3 Experimental settings

The distribution of POS n-grams in a collection and the relation between PIS_1 and PIS_2 to IDF are studied in five standard TREC collections using non-biased experimental settings. The experimental settings consist of:

- the POS tagger used to tag the collection, presented in Section 6.3.1;
- the collection characteristics (size, domain), presented in Section 6.3.2; and
- the order n of POS n-grams, presented in Section 6.3.3.

Each of these settings is presented separately in order to show that it does not bias this study.

6.3.1 Part of speech tagger

Chapter 3 presented three standard POS taggers (Mihalcea, 2003):

- the Transformation Based (Brill) tagger (Brill, 1995), described in Section 3.4.1;
- the Maximum Entropy (Mxpost) tagger (Ratnaparkhi, 1996), described in Section 3.4.2; and
- the TreeTagger (Schmid, 1994), described in Section 3.4.3.

The aim of this section is to compare these three taggers, and select a POS tagger that is most appropriate for these experiments. The selected POS tagger will be used for the rest of the thesis.

Table 6.1 shows the functionalities of these three POS taggers, (*tokenisation* and *HTML-like processing*), and their reported POS tagging accuracy.

Tokenisation refers to splitting text into individual tokens. The Brill tagger and Mxpost require their input to be pre-tokenised because they do not include a tokeniser. This means that an extra pre-processing step must be added before using them for POS tagging. The TreeTagger also requires input to be tokenised, but includes a tokeniser, so no extra pre-processing is needed. This is advantageous.

Features	**Brill tagger**	**Mxpost tagger**	**TreeTagger**
tokenisation	not included	not included	included
HTML-like processing	not included	not included	included
accuracy on WSJ	96.5%	96.6%	96.4%

Table 6.1: Functionalities of the Brill tagger, the Mxpost tagger, and the TreeTagger.

HTML-like processing refers to the processing of specific annotation such as HTML tags, i.e., not tagging this annotation as if it were text, but ignoring it. The Brill tagger and Mxpost cannot ignore HTML-like annotation. This means than an extra pre-procesing step is required to remove annotation, and an extra post-processing step is required to re-apply the annotation. HTML-like annotation needs to be re-applied when using standard TREC collections for instance, because it marks document identification and structure, needed for retrieval. Unlike the Brill tagger and Mxpost, the TreeTagger has an option to ignore HTML-like annotation. This is advantageous.

The Brill tagger, Mxpost, and the TreeTagger are approximately equally accurate (Mihalcea, 2003). For English, POS taggers are typically evaluated on the Penn Treebank Wall Street Journal (WSJ) corpus, which is a human POS annotated corpus, containing 4.5 million words of American English (Marcus *et al.*, 1993). All three POS taggers report very similar accuracy levels on WSJ, as shown in Table 6.1. Hence, these POS taggers are approximately equally accurate.

To verifiy further the accuracy of these POS taggers, a small experiment is conducted: a collection is POS tagged with each of these three POS taggers separately, and the relative frequency (RF) of individual parts of speech in each of the three POS tagged versions of the collection is compared. (The relative frequency was introduced in Section 4.2.1, Equation 4.1, page 51.) Ideally, all POS tags should have the same RF, because they belong to the same text, but, in practice, using different POS taggers can produce slightly different outputs. The more similar the relative frequency of a part of speech is across the three POS tagged versions of the collection, the more similar the performance of the POS taggers is to each other.

The POS tagged collection is the Associated Press (AP) collection, which contains newswire text. The AP is one of the five TREC collections used in this thesis. All five collections are presented in the next section, in Table 6.3. As stated in Section 3.2, page 38, the POS tags used are the ones corresponding to

POS tagging AP with 3 different taggers				
pos	Brill tagger	Mxpost tagger	TreeTagger	st. deviation
NN	0.35	0.34	0.34	0.005
VB	0.11	0.10	0.11	0.005
JJ	0.06	0.07	0.07	0.005
VR	0.03	0.03	0.04	0.005
RB	0.02	0.03	0.03	0.005
IN	0.14	0.14	0.14	none
DT	0.10	0.09	0.10	0.005
MD	0.05	0.05	0.05	none
PP	0.05	0.05	0.04	0.005
CD	0.05	0.04	0.04	0.005
CC	0.03	0.03	0.03	none
PO	0.03	0.02	0.01	0.008

Table 6.2: Relative frequency of parts of speech in AP.

the primary grammatical categories of the Penn TreeBank tagset (presented in Table 3.1, page 40).

Table 6.2 presents the relative frequency of individual parts of speech in three differently tagged versions of the AP collection. For each POS tag, its relative frequency values in the different versions of the collection are very similar to each other. Hence, the standard deviation values reported are very low (between 0 - 0.008, with an average of 0.003). POS tagging the AP with any of the three POS taggers does not bias the relative frequency of any part of speech. This finding is expected, because all three POS taggers are reported to be approximately equally accurate.

The conclusion is that none of the three POS taggers introduces bias. The TreeTagger is more appealing, because it has extra functionalities (a tokeniser and HTML-like processing). For these reasons, the TreeTagger is used in the rest of the thesis.

6.3.2 Collection characteristics

The second parameter in these experiments is the characteristics of the collections used. Five different standard TREC test collections are used: AP, Disks 4&5, WT2G, WT10G, and .GOV. Table 6.3 presents each collection. AP and Disks 4&5 contain news releases from printed media; these collections are mostly homogeneous (they contain documents from a single source). WT2G, WT10G, and .GOV consist of crawled pages from the Web, which is itself a heterogeneous source (albeit from a restricted .gov domain in the case of .GOV). These five

Collection	Size	Documents	Unique Terms	Domain
AP	742 MB	242,918	315,539	journalistics
Disks 4&5	1.9 GB	528,155	521,469	journalistic
WT2G	2 GB	247,491	1,002,586	Web
WT10G	10 GB	1,692,096	3,140,837	Web
.GOV	18.1 GB	1,247,753	2,788,457	Web (.gov domain)

Table 6.3: Characteristics of the AP, Disks 4&5, WT2G, WT10G, and .GOV collections.

collections also differ in their word statistics, since larger collections do not necessarily contain more unique terms. For instance, even though .GOV is roughly 8 times larger than WT10G, WT10G contains many more unique terms that .GOV. Overall, the collections vary in size (742MB - 18.1GB), word statistics, and domain (newswire, Web).

To test the effect of collection characteristics on the distribution of POS n-grams and the estimation of PIS, a small experiment is conducted: The collections are POS tagged, and two things are observed:

1. Similarly to the experiment in Section 6.3.1, the relative frequency of individual parts of speech is compared across collections (Section 6.3.2.1).

2. For each collection separately, POS n-grams are extracted and ranked by their frequency in the collection. Then, the ranked lists are compared across collections (Section 6.3.2.2).

The more similar (i) the relative frequencies of parts of speech, and (ii) the ranked lists of POS n-grams, across collections, the smaller the effect of the collection characteristics towards specific parts of speech or POS n-grams. Hence, the smaller the effect of collection characteristics on the distribution of POS n-grams and on the computation of PIS.

Next, these two observations are discussed separately.

6.3.2.1 Individual parts of speech

Five collections are POS tagged. Separately for each collection, the relative frequency (RF) of individual parts of speech is computed, using Equation 4.1, page 51. This is similar to the experiment in Section 6.3.1, with one difference: in Section 6.3.1, the RF of individual parts of speech was compared in versions

Relative frequency of parts of speech in five TREC collections							
POS	AP	Disks 4&5	WT2G	WT10G	.GOV	st. deviation	mean
NN	0.34	0.34	0.40	0.39	0.43	0.035	0.38
VB	0.11	0.11	0.10	0.10	0.07	0.015	0.10
JJ	0.07	0.07	0.08	0.07	0.08	0.005	0.07
VR	0.04	0.04	0.03	0.03	0.03	0.005	0.03
RB	0.03	0.04	0.03	0.03	0.02	0.006	0.03
IN	0.14	0.12	0.09	0.10	0.11	0.017	0.11
DT	0.10	0.09	0.07	0.08	0.08	0.010	0.08
MD	0.05	0.07	0.06	0.06	0.03	0.013	0.05
PP	0.04	0.05	0.05	0.05	0.01	0.015	0.04
CD	0.04	0.03	0.04	0.05	0.09	0.021	0.05
CC	0.03	0.03	0.03	0.03	0.03	none	0.03
PO	0.01	0.01	<0.01	<0.01	<0.01	0.005	<0.01

Table 6.4: Relative frequency of parts of speech (AP, Disks 4&5, WT2G, WT10G, .GOV).

of the same collection that were tagged differently, whereas now the RF of individual parts of speech is compared in different collections that have been tagged identically. Hence, RF scores are expected to be less similar to each other here, compared to the ones in Section 6.3.1.

Table 6.4 shows the RF values of individual parts of speech in each collection. The standard deviation of RF of all parts of speech is very low (between 0 - 0.035, with an average of approximately 0.01). Hence, none of these collections is particularly biased to a specific part of speech.

The conclusion is that the different collection characteristics (size, domain) do not seem to affect the distribution of individual parts of speech per collection.

6.3.2.2 Part of speech n-grams

For each collection separately, POS n-grams are extracted and ranked by their frequency in the whole collection. This gives five lists of POS n-grams, one for each collection. The similarity between these pairs of lists is compared. The more similar they are, the less biased either of the collections is for a POS n-gram. This is repeated for all possible pairs of the five collections. The order n is varied between 4-5. (Setting n is discussed in Section 6.3.3, where it is shown that n = 4-5 is an unbiased setting.) Tables 6.5 - 6.6 show Spearman's ρ correlation values of POS n-grams for each collection pair. All collection pairs are very strongly correlated for n=4,5. This means that different collections, of different

Spearman's rank correlation (ρ) for ranks of POS 4-grams

	AP	Disks 4&5	WT2G	WT10G	.GOV
AP	-	$\rho \approx 1.0$	$\rho \approx 1.0$	$\rho \approx 1.0$	$\rho \approx 1.0$
Disks 4&5	$\rho \approx 1.0$	-	$\rho \approx 1.0$	$\rho \approx 1.0$	$\rho \approx 1.0$
WT2G	$\rho \approx 1.0$	$\rho \approx 1.0$	-	$\rho \approx 1.0$	$\rho \approx 1.0$
WT10G	$\rho \approx 1.0$	$\rho \approx 1.0$	$\rho \approx 1.0$	-	$\rho \approx 1.0$
.GOV	$\rho \approx 1.0$	$\rho \approx 1.0$	$\rho \approx 1.0$	$\rho \approx 1.0$	-

Table 6.5: Spearman's rank correlation between ranks of POS 4-grams (AP, Disks 4&5, WT2G, WT10G, .GOV).

Spearman's rank correlation (ρ) for ranks of POS 5-grams

	AP	Disks 4&5	WT2G	WT10G	.GOV
AP	-	$\rho \approx 1.0$	$\rho \approx 1.0$	$\rho \approx 1.0$	$\rho \approx 1.0$
Disks 4&5	$\rho \approx 1.0$	-	$\rho \approx 1.0$	$\rho \approx 1.0$	$\rho \approx 1.0$
WT2G	$\rho \approx 1.0$	$\rho \approx 1.0$	-	$\rho \approx 1.0$	$\rho \approx 1.0$
WT10G	$\rho \approx 1.0$	$\rho \approx 1.0$	$\rho \approx 1.0$	-	$\rho \approx 1.0$
.GOV	$\rho \approx 1.0$	$\rho \approx 1.0$	$\rho \approx 1.0$	$\rho \approx 1.0$	-

Table 6.6: Spearman's rank correlation between ranks of POS 5-grams (AP, Disks 4&5, WT2G, WT10G, .GOV).

size and domain, produce very similar lists of POS n-grams. This is not entirely surprising because, as Table 6.4 showed, the five collections contain very similar proportions of individual parts of speech. Since the possible arrangements of these parts of speech are bounded by the same grammatical rules, their resulting POS n-grams are likely to be recurrent. However, a possible factor for the very strong correlations reported in Tables 6.5 - 6.6 could also be due to an extent to the POS tagging process, which tends to 'favour' popular POS tag arrangements[1]. More simply, the more frequently a POS n-gram occurs, the more likely it is to occur again, especially for collections of such large size (742MB - 18.1GB), and for such a small number of POS tags (14 classes).

Figures 6.1 - 6.2 plot the distribution of POS 4-grams in each collection separately. The x axis is the frequency of a POS n-gram in the collection. The y axis is the rank of that frequency. The corresponding plots for POS 5-grams are included in Appendix D, Figures B.1 - B.2, pages 156 - 157. These figures show that the distribution of POS n-grams is similar, not only across collections, but also for n=4-5. This point about POS n-gram order is discussed in Section 6.3.3. The dis-

[1] This is common practice for most POS taggers that use statistics.

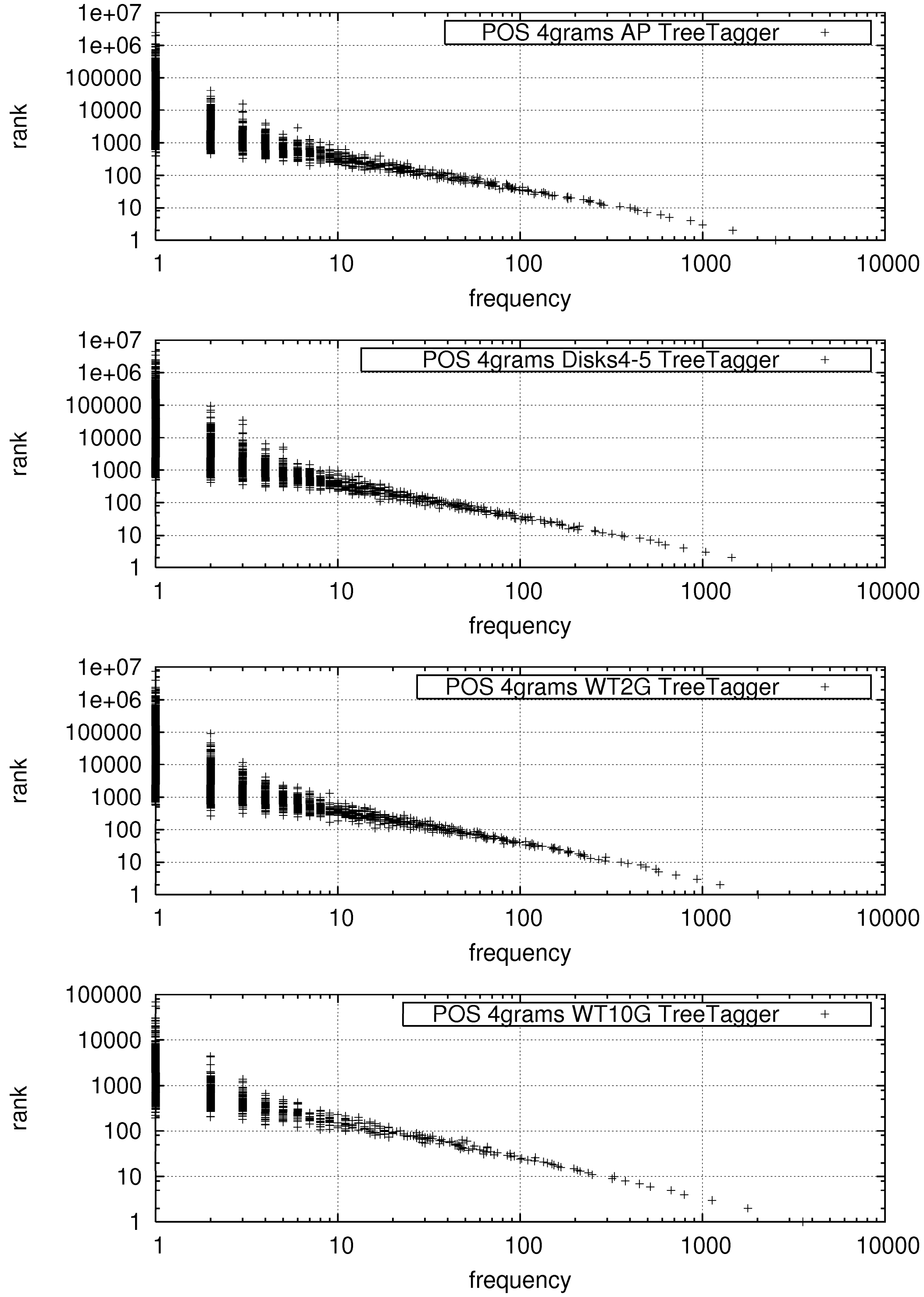


Figure 6.1: Distribution of POS 4-grams (AP, Disks 4&5, WT2G, WT10G).

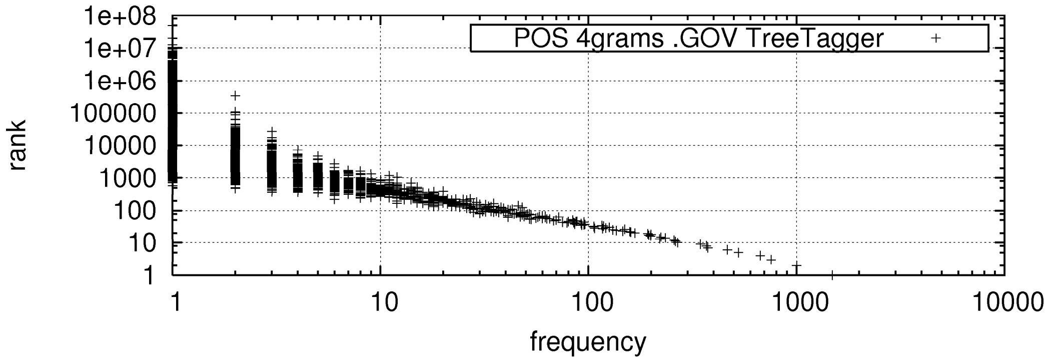


Figure 6.2: Distribution of POS 4-grams (.GOV).

tribution of POS 4-grams and 5-grams across collections resembles a *large number of rare events* distribution, a type of which are *power laws* (Mitzenmacher, 2004; Newman, 2005). As the name suggests, in these distributions there are many very rare events, and few very common events. As stated in Section 2.3.2.2, Zipf showed that in language, words are distributed in this way (Li, 1992; Sigurd *et al.*, 2004; Zipf, 1949). For the purposes of this experiment, the fact that POS 4-grams and 5-grams are distributed similarly to power laws across the five collections indicates that their distribution tends to be regular across collections, hence not heavily dependent on collection characteristics (Naranan & Balasubrahmanyan, 1998).

The findings from Tables 6.5 - 6.6, Figures 6.1 - 6.2, and Figures B.1 - B.2 are also illustrated in Tables 6.7 and 6.8, which show the ten most frequent POS 4-grams and POS 5-grams in each collection. POS n-grams common to all collections, in the top ten, are in bold. Bold with asterisk * denotes POS n-grams common to four out of five collections, in the top ten. Most POS n-grams among the top ten are common across collections. An exception to this is the series of cardinal numbers (CD) observed in .GOV, which indicates that .GOV contains a considerably larger proportion of numbers, than any other collection. This is the type of domain-specific characteristic that n-gram based language models use to represent language regularities in applications like text classification (presented in Section 4.2.2).

Ten most frequent POS 4-grams in the collection (in decreasing order)				
AP	Disks 4&5	WT2G	WT10G	.GOV
NN NN NN NN	**NN IN DT NN**	**NN NN NN NN**	**NN NN NN NN**	**NN NN NN NN**
NN IN DT NN	**NN NN NN NN**	**NN IN DT NN**	**NN IN DT NN**	**NN IN DT NN**
IN DT NN NN	**IN DT NN IN**	**NN IN NN NN**	**NN IN NN NN**	**NN IN NN NN**
NN IN NN NN	**IN DT JJ NN**	**IN DT NN IN**	**IN DT NN IN**	**IN DT NN NN**
IN DT JJ NN	**IN DT NN NN**	**IN DT JJ NN**	**IN DT NN NN**	CD JJ CD CD
IN DT NN IN	**NN IN NN NN**	**IN DT NN NN**	**IN DT JJ NN**	CD CD JJ CD
NN NN IN NN *	DT NN IN DT	NN IN JJ NN	DT NN IN DT	**IN DT NN IN**
DT NN IN NN *	DT JJ NN IN	**DT NN IN NN** *	**DT NN IN NN** *	CD CD CD CD
DT JJ NN IN	**DT NN IN NN** *	**NN NN IN NN** *	**NN NN IN NN** *	**NN NN IN NN** *
DT NN IN DT	NN IN DT JJ	DT JJ NN IN	NN IN JJ NN	**IN DT JJ NN**

Table 6.7: Ten most frequent POS 4-grams (AP, Disks 4&5, WT2G, WT10G, .GOV).

Ten most frequent POS 5-grams in the collection (in decreasing order)				
AP	Disks 4&5	WT2G	WT10G	.GOV
NN NN NN NN NN	**NN NN NN NN NN**	**NN NN NN NN NN**	**NN NN NN NN NN**	**NN NN NN NN NN**
NN IN DT NN NN	**NN IN DT NN NN**	**NN IN DT NN NN**	**NN IN DT NN NN**	CD CD JJ CD CD
NN IN DT JJ NN	**DT NN IN DT NN**	**DT NN IN DT NN**	**DT NN IN DT NN**	CD JJ CD CD JJ
DT NN IN DT NN	**NN IN DT JJ NN**	**NN IN DT NN IN**	**NN IN DT NN IN**	JJ CD CD JJ CD
NN IN DT NN IN	**NN IN DT NN IN**	**NN IN DT JJ NN**	**NN IN DT JJ NN**	CD CD CD CD CD
JJ NN IN DT NN *	**JJ NN IN DT NN** *	**JJ NN IN DT NN** *	**NN IN NN NN NN** *	**NN IN DT NN NN**
NN IN NN NN NN *	IN DT NN IN DT	**NN IN NN NN NN** *	**JJ NN IN DT NN** *	**DT NN IN DT NN**
NN NN IN DT NN *	IN DT NN IN NN *	**IN DT NN IN NN** *	**NN NN IN NN NN**	**NN IN DT NN IN**
IN DT NN NN NN	**NN NN IN DT NN** *	**NN NN IN DT NN** *	**NN NN IN DT NN** *	**NN IN NN NN NN** *
IN DT NN IN NN *	IN DT JJ NN IN	IN DT NN IN DT	**IN DT NN IN NN** *	**NN IN DT JJ NN**

Table 6.8: Ten most frequent POS 5-grams (AP, Disks 4&5, WT2G, WT10G, .GOV).

Overall, Tables 6.7 and 6.8 confirm the conclusions of Tables 6.5 - 6.6, Figures 6.1 - 6.2 and Figures B.1 - B.2, that the distribution of POS n-grams across collections does not alter much. It is concluded that collection characteristics do not bias the extraction of POS n-grams.

6.3.3 Order n of part of speech n-grams

The order of the POS n-grams n is the last parameter in these experiments. Generally in n-grams, n should be large enough to encode contextual information, and small enough to allow for sensible estimations, as discussed in Section 4.2.1.

In this study, the most appropriate order of POS n-grams needs to be selected. Appropriate here means that it should give enough POS n-grams to derive PIS from, and also be able to capture linguistic structure to an extent. To make this decision, a small experiment is conducted: POS n-grams are extracted from a collection, varying n between 1-100. For each n, the total number of POS n-grams extracted is counted. E.g., for n=1, there will be a total of 14 POS 1-grams, one for each part of speech. The n that gives more POS n-grams, and also which captures linguistic structure to some extent is more appropriate here. Generally in n-grams, as n increases, n-gram frequency tends to decrease, so the selection of n is usually a compromise between the width of the context captured (n) and n-gram frequency.

For this experiment, the AP collection is used. (Collection characteristics are shown in Table 6.3, page 80.) The order n is ranged within [1,100] with an increasing interval:

- from 1 to 15, with an interval of 1,
- from 20 to 100, with an interval of 5.

Figure 6.3 plots n (x axis) against the frequency of POS n-grams in the collection (y axis) for n = 1-100, from the AP collection. n = 4-7 gives more POS n-grams (>5000), with n=4 and n=7 giving roughly the same number of POS n-grams ($\approx$ 5000).

From Figure 6.3 it is concluded that n = 4-7 seems to be more appropriate for this study, because it gives the most POS n-grams, which also capture linguistic structure between 4 and 7 words.

A small additional experiment is conducted to look at how POS n-gram order n affects the distribution of POS n-grams in the collection. The aim is to make sure that the selected value for n, not only gives many POS n-grams, but also

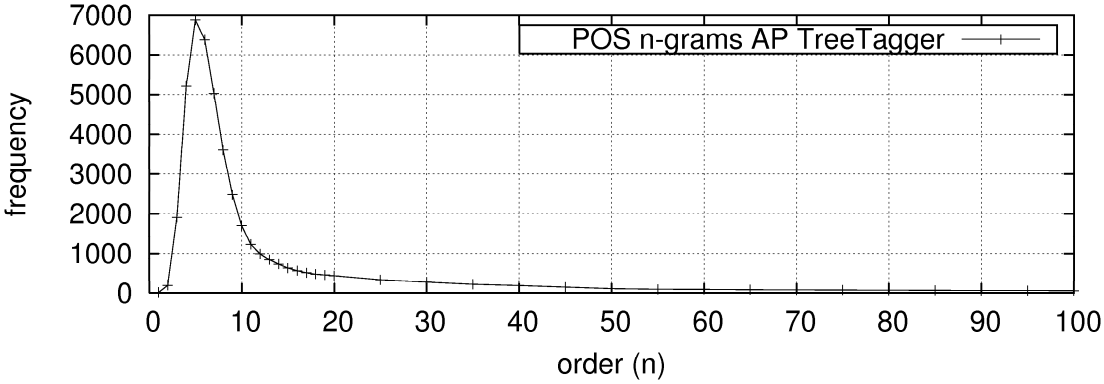


Figure 6.3: Order n value versus number of unique POS n-grams (AP).

corresponds to a POS n-gram distribution that is generally representative (i.e., one that does not give irregular distributions). Similarly to the experiments in Section 6.3.2.2, where frequency - rank plots of POS n-grams were presented across different collections, here the frequency - rank plot of POS n-grams from the AP collection is presented, while varying n between 1 and 100. Figure 6.4 shows the resulting plots for n = 4-7. The corresponding plots for all other n values used are included in Appendix B, Figures B.3 - B.10, page 155.

Figure 6.4 shows that the distribution of POS n-grams in a collection does not alter much when varying n between 4-7. It also shows that the distribution of POS n-grams resembles a power law distribution, characterised by many POS n-grams of low frequency, and a few of very high frequency.

Figures B.3 - B.10 in Appendix B, page 155, show that the shape of the plot for POS n-grams is preserved for n between 4-40 (Figures B.3 - B.7, pages 158 - 162). n=1,2 do not provide enough POS n-grams to make a distribution, since, if all frequencies are unique, all frequency ranks are 1 (Figure B.3, page 158). For n >15 POS n-grams are increasingly sparse (Figures B.6 - B.10, pages 161 - 165).

The conclusion is that setting n=4-7 gives POS n-grams that are representative of how POS n-grams are distributed generally in language, and also gives the most number of POS n-grams. For these reasons, POS 4-grams will be used in the rest of this thesis.

6.3.4 Summary of experimental settings

Three parameters are involved in extracting POS n-grams from collections: (i) the POS tagger used to POS tag the collections, (ii) the collection characteristics

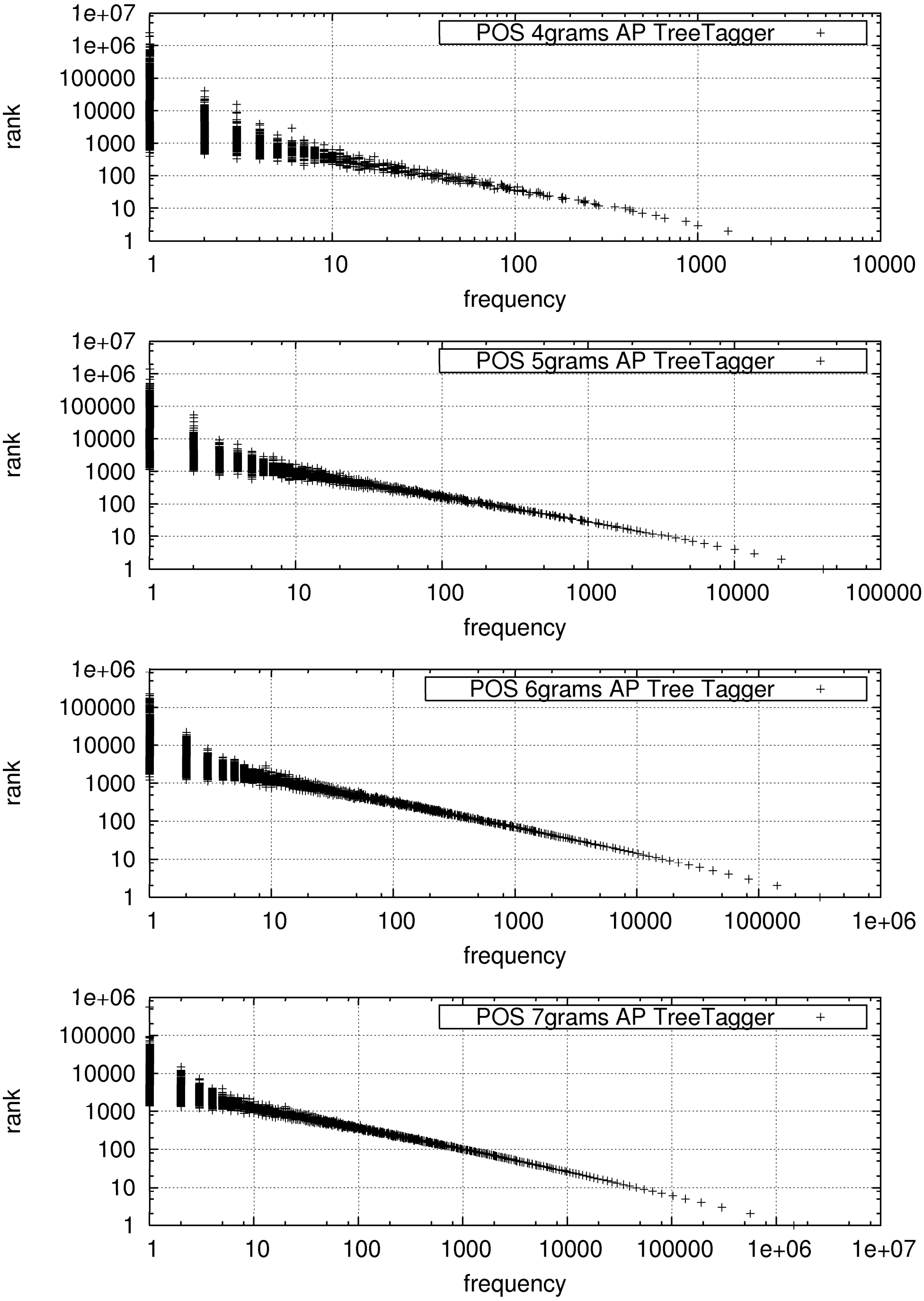


Figure 6.4: Distribution of POS 4-grams, 5-grams, 6-grams and 7-grams (AP).

from which POS n-grams are extracted, and (iii) the order n of the POS n-grams. In Section 6.3, three standard POS taggers were compared and found to have no considerable impact on POS n-gram distribution, because the taggers are approximately equally accurate. Five collections of different characteristics were compared and found to have no considerable impact on POS n-gram distribution, because POS occurrence is generally robust across collections. The order n of POS n-grams was ranged between 1-100 and 4-7 was found to be the value that gives the most POS n-grams that also captures linguistic structure to an extent. It was also found that varying n between 4-7 has no considerable impact on POS n-gram distribution. The conclusion is that, when studying POS n-grams, using the TreeTagger and setting n=4 are unbiased settings, and also that the overall study is not expected to be considerably collection-dependent.

In addition, Section 6.3 showed that the distribution of POS n-grams in a collection resembles a power law, similarly to the distribution of words, and also that it tends to be similar across collections. The remainder of this chapter looks at whether the term information score derived from POS n-grams (PIS) is correlated to IDF.

6.4 Experimental methodology

In order to test if PIS_1 and PIS_2 are correlated to IDF, PIS_1 and PIS_2 are compared to IDF as follows:

1. The IDF, PIS_1 and PIS_2 of all terms[1] in a collection is computed;

2. The correlation between the IDF:PIS_1 and IDF:PIS_2 of all the terms in a collection is measured.

The process is repeated separately for five standard TREC collections. Each collection is processed twice separately, once to compute IDF, and once to compute PIS_1 and PIS_2.

Prior to computing IDF, standard IR pre-processing operations apply (presented in Section 2.3): terms are tokenised on whitespace and punctuation marks, and lower-cased. Stopwords are removed, and words are stemmed with the Porter stemming algorithm (Porter, 1980). The above process is done using the Terrier IR platform (Ounis *et al.*, 2007). IDF is computed with Equation 2.1, page 16.

[1] More accurately, all indexed terms in a collection.

Prior to computing PIS_1 and PIS_2, each collection is POS tagged with the TreeTagger, which includes a tokeniser (presented in Section 3.4.3). POS 4-grams are extracted (setting n=4 is discussed in Section 6.3.3). PIS_1 is computed with Equation 5.14, page 69. PIS_2 is computed with Equation 5.21, page 73. Both Equations 5.14 and 5.21 include the variables λ and ϱ, which represent the probability that a first and second degree part of speech is informative, respectively (presented in Section 5.3.2). For these experiments, the values of λ and ϱ are derived using Bayes Rule, as shown in Section 5.3.2.2, by setting $\lambda = 1$ and solving for ϱ. In Chapter 7, it is shown that $\lambda = 1$ is the optimal value for λ, when used as part of an IR system and tuned for retrieval perfrormance.

In order to test whether the computations of PIS_1 and PIS_2 are collection-dependent, two rounds of experiments are defined:

- **Round 1:** Compare IDF:PIS_1 & IDF:PIS_2, having computed PIS_1 and PIS_2 from the same collection. This is referred to as **per-collection comparison** and is presented in Section 6.5.1.

- **Round 2:** Compare IDF:PIS_1 & IDF:PIS_2, having computed PIS_1 and PIS_2 from a different collection. For example, given two collections, $C1$ and $C2$, and a term t occurring in $C1$, the PIS_1 (resp. PIS_2) of that term is computed using POS n-grams from $C2$. This is referred to as **cross-collection comparison** and is presented in Section 6.5.2.

Conducting the above experiments on all five collections would result in 2×5^5 possible combinations. This large number of combinations is not necessary, because all collections used are standard TREC datasets, hence they are expected to be large enough and representative enough of language use in the journalistic and general Web domain. For this reason, the five collections are split evenly into two sets, and round 1 is realised with one set, and round 2 with the other set. Each set contains three collections; one collection is common to both sets. The collections are split so that both sets contain a 'balanced' number of unique terms: Set 1 includes Disks 4&5, WT2G, and WT10G, with a total of 4.7M unique terms. Set 2 contains AP, WT2G, and .GOV, with a total of 4.1M unique terms. Collection statistics are displayed in Table 6.3, page 80. Table 6.9 displays all the collection combinations used for the per-collection and cross-collection comparisons presented next: IDF, PIS_1 & PIS_2 are computed for terms in the collections in the first column; PIS_1 & PIS_2 are computed using POS n-grams extracted from the collections in the second column.

Per-collection combinations: IDF:PIS$_1$ & IDF:PIS$_2$	
IDF:PIS$_1$, IDF:PIS$_2$ for terms in Disks 4&5	PIS$_1$ & PIS$_2$ computed from Disks 4&5
IDF:PIS$_1$, IDF:PIS$_2$ for terms in WT2G	PIS$_1$ & PIS$_2$ computed from WT2G
IDF:PIS$_1$, IDF:PIS$_2$ for terms in WT10G	PIS$_1$ & PIS$_2$ computed from WT10G
Cross-collection combinations: IDF:PIS$_1$ & IDF:PIS$_2$	
IDF:PIS$_1$, IDF:PIS$_2$ for terms in AP	PIS$_1$ & PIS$_2$ computed from Disks 4&5
	PIS$_1$ & PIS$_2$ computed from WT2G
	PIS$_1$ & PIS$_2$ computed from WT10G
IDF:PIS$_1$, IDF:PIS$_2$ for terms in WT2G	PIS$_1$ & PIS$_2$ computed from Disks 4&5
	PIS$_1$ & PIS$_2$ computed from WT10G
IDF:PIS$_1$, IDF:PIS$_2$ for terms in .GOV	PIS$_1$ & PIS$_2$ computed from Disks 4&5
	PIS$_1$ & PIS$_2$ computed from WT2G
	PIS$_1$ & PIS$_2$ computed from WT10G

Table 6.9: Collection combinations of IDF, PIS1 and PIS$_2$.

Spearman's rank correlation ρ		
Collection	IDF versus PIS_1	IDF versus PIS_2
Disks 4&5	ρ=0.402	ρ=0.945
WT2G	ρ=0.440	ρ=0.955
WT10G	ρ=0.622	ρ=0.990

Table 6.10: Spearman's rank correlation between IDF and PIS (Disks 4&5, WT2G, WT10G).

Because PIS_1 and PIS_2 are derived from POS n-grams, and, Section 6.3.2 showed that collection statistics do not affect considerably the distribution of POS n-grams, it is expected that IDF:PIS_1 and IDF:PIS_2 are correlated similarly across collections in both rounds of experiments.

6.5 Experimental results

6.5.1 Per-collection comparison

The aim is to compare the IDF:PIS_1 and IDF:PIS_2 for each term in a collection, having computed PIS_1 and PIS_2 from the same collection. To do so, the following steps are taken:

- **Step 1.** Compute the IDF for each term in a collection, and sort the IDF values of all the terms in the collection.
- **Step 2.** Compute the PIS_1 for each term in the same collection, and sort the PIS values of all the terms in the collection. Do the same for PIS_2 separately.
- **Step 3.** Compute the correlation between the two sorted lists (IDF:PIS_1, IDF:PIS_2), using Spearman's rank coefficient.

This is repeated separately for Disks 4&5, WT2G, and WT10G. These collections make the first set of collections, as discussed in Section 6.4.

The results are presented in Table 6.10. Table 6.10 displays Spearman's ρ for each IDF:PIS_1 and IDF:PIS_2 pair. Table 6.10 shows two things:

- for each collection, all IDF:PIS_1 and IDF:PIS_2 combinations are positively correlated (between ρ= 0.402 - 0.990); and

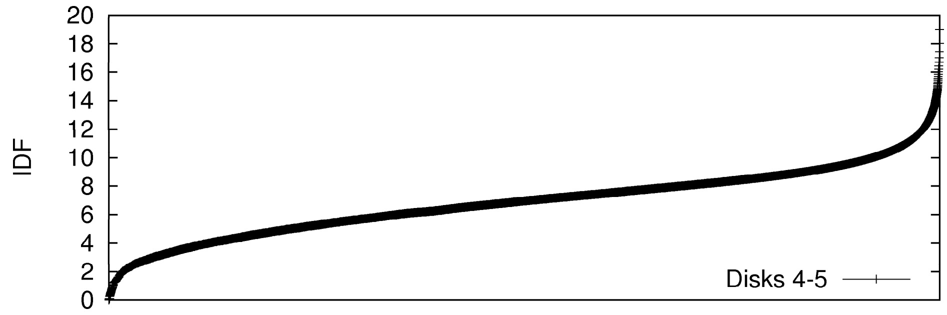


Figure 6.5: IDF of terms in Disks 4&5.

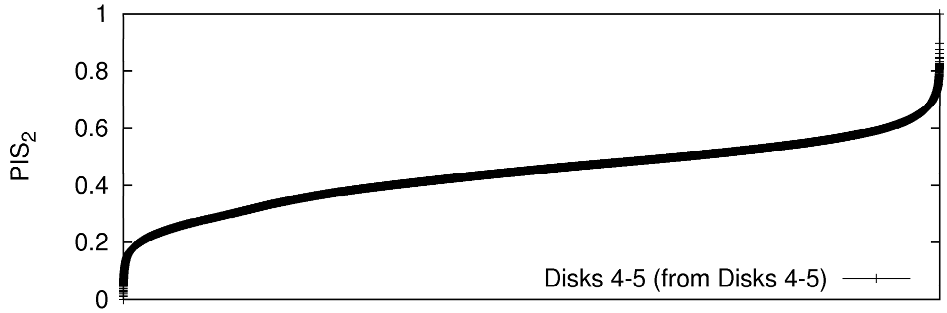


Figure 6.6: PIS_2 of terms in Disks 4&5.

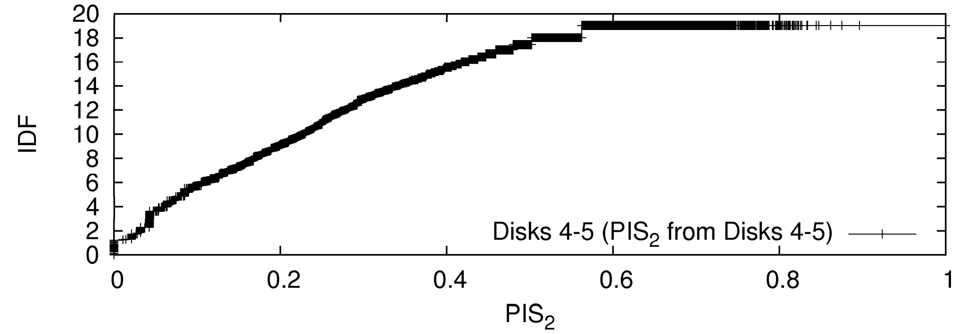


Figure 6.7: PIS_2 versus IDF of terms in Disks 4&5.

- for each collection, IDF:PIS$_1$ is less correlated than IDF:PIS$_2$ (between ρ=0.402 - 0.622 for IDF:PIS$_1$, and between ρ=0.945 - 0.990 for IDF:PIS$_2$).

Finding a very high correlation between IDF:PIS$_2$ is expected, because the computation of both PIS$_2$ and IDF consider term frequency in a collection (in the case of PIS$_2$ as an 'inflated term frequency' as discussed in Section 5.3.4.4, and in the case of IDF as the number of documents that contain the term). This point is also illustrated in Figure 6.5.1 for the Disks 4&5 collection as follows: Figure 6.5.1 plots the IDF and PIS$_2$ of all terms in the collection, first separately, and then against each other. Figure 6.5.1 shows that the plot of IDF has a very similar shape to the PIS$_2$ plot, for Disks 4&5, and also that the plot of IDF against PIS$_2$ approximates a straight line. This is also the case for WT2G and WT10G, the corresponding plots of which are included in Appendix D, Figures D.1 - D.6, pages 169 - 170.

Overall, Table 6.10 shows that IDF:PIS$_1$ and IDF:PIS$_2$ are correlated, when PIS$_1$ and PIS$_2$ are computed from the same collection. The next section shows that PIS$_1$ and PIS$_2$ are correlated to IDF, even when PIS$_1$ and PIS$_2$ are computed from a different collection.

6.5.2 Cross-collection comparison

The aim is to compare the IDF:PIS$_1$ and IDF:PIS$_2$ for each term in a collection, having computed PIS$_1$ and PIS$_2$ from a different collection. This will confirm that PIS$_1$ and PIS$_2$ are collection-independent. To do so, Steps 1 - 3 described in Section 6.5.1, are repeated with one difference: when computing PIS$_1$ and PIS$_2$, POS n-grams from another collection are used. The collections used to compute IDF are referred to as *idf collections*. The collections used to compute PIS$_1$ and PIS$_2$ are referred to as *pis collections*. The *idf collections* are AP, WT2G, and .GOV. This is the second set of collections, presented in Section 6.4. The *pis collections* are Disks 4&5, WT2G, and WT10G. This is the first set of collections, presented in Section 6.4. The possible combinations are summarised in Table 6.9. The combination WT2G - WT2G is omitted from this section, because it was already presented in Section 6.5.1.

Similarly to Section 6.5.1, Spearman rank correlations between the IDF-PIS$_1$ and PIS$_2$ combinations are shown in Table 6.11.

Table 6.11 displays Spearman's ρ for each IDF:PIS$_1$ and IDF:PIS$_2$ combination. The collections in brackets are the collections from which POS n-grams were

Spearman's rank correlation (ρ)		
collections	IDF:PIS$_1$	IDF:PIS$_2$
AP (PIS$_1$ and PIS$_2$ computed from Disks 4&5)	ρ=0.499	ρ=0.968
AP (PIS$_1$ and PIS$_2$ computed from WT2G)	ρ=0.402	ρ=0.955
AP (PIS$_1$ and PIS$_2$ computed from WT10G)	ρ=0.580	ρ=0.990
WT2G (PIS$_1$ and PIS$_2$ computed from Disks 4&5)	ρ=0.623	ρ=0.998
WT2G (PIS$_1$ and PIS$_2$ computed from WT10G)	ρ=0.616	ρ=0.990
.GOV (PIS$_1$ and PIS$_2$ computed from Disks 4&5)	ρ=0.628	ρ=1.000
.GOV (PIS$_1$ and PIS$_2$ computed from WT2G)	ρ=0.400	ρ=0.993
.GOV (PIS$_1$ and PIS$_2$ computed from WT10G)	ρ=0.590	ρ=0.998

Table 6.11: Spearman's rank correlation between IDF and PIS (AP, WT2G, .GOV).

extracted and used to compute the two versions of PIS. The conclusions drawn from Table 6.11 are similar to the conclusions drawn from Table 6.10, namely:

- for each collection, all IDF:PIS$_1$ and IDF:PIS$_2$ combinations are positively correlated (between ρ= 0.400 - 1.000); and

- for each collection, IDF:PIS$_1$ is less correlated than IDF:PIS$_2$ (between ρ=0.400 - 0.628 for IDF - PIS$_1$, and between ρ=0.955 - 1.000 for IDF:PIS$_2$).

As expected, IDF:PIS$_2$ are strongly correlated (this point was discussed in Section 6.5.1). This point is also illustrated in Figures 6.8 - 6.14 for the AP collection as follows: Figure 6.8 plots the IDF of all terms in the collection; Figures 6.9 - 6.11 plot the PIS$_2$ of all terms in the collection, where PIS$_2$ has been computed using POS n-grams from a different collection than AP, namely Disks 4&5, WT2G, and WT10G. Figures 6.12 - 6.14 plot each of these plots of PIS$_2$ against the IDF of all terms in AP, and show that the plot of IDF has a very similar shape to the PIS$_2$ plot, for AP, and also that the plot of IDF against PIS$_2$ approximates a straight line. This is also the case for WT2G and .GOV, the corresponding plots of which are included in Appendix D, Figures D.7 - D.17, pages 171 - 174.

Overall, Section 6.5.2 shows that IDF:PIS$_1$ and IDF:PIS$_2$ are correlated, when PIS$_1$ and PIS$_2$ are computed from any collection. This means that the computation of PIS$_1$ and PIS$_2$ is indeed collection-independent. It is concluded that PIS$_1$ is correlated to IDF and that PIS$_2$ is strongly correlated to IDF. Also, the computation of PIS$_1$ and PIS$_2$ is not collection-dependent. This is validated across five standard TREC collections.

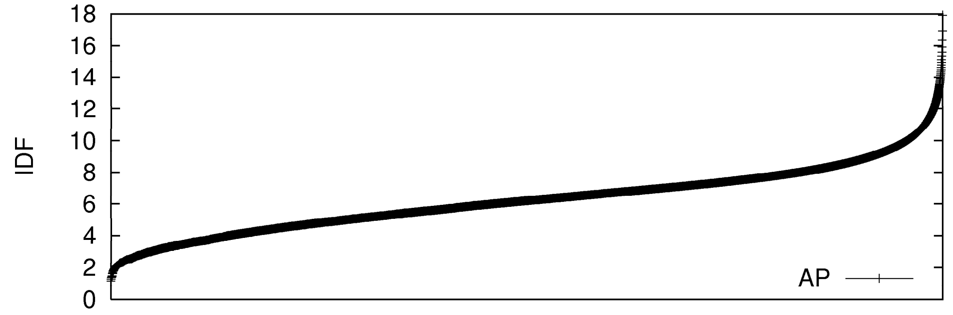


Figure 6.8: IDF of terms in AP.

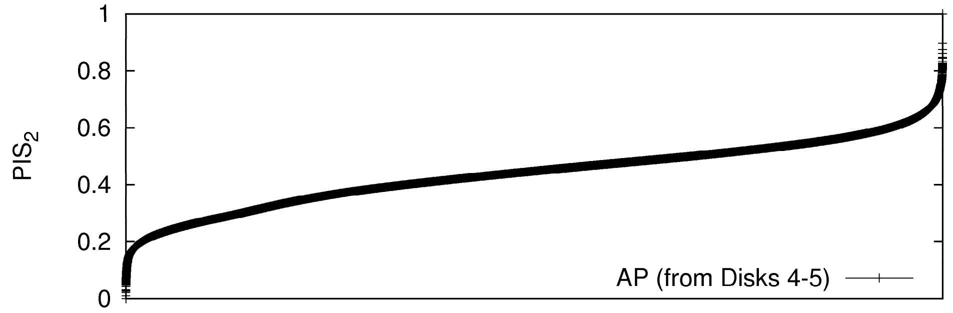


Figure 6.9: PIS_2 of terms in AP (POS 4-grams from Disks 4&5).

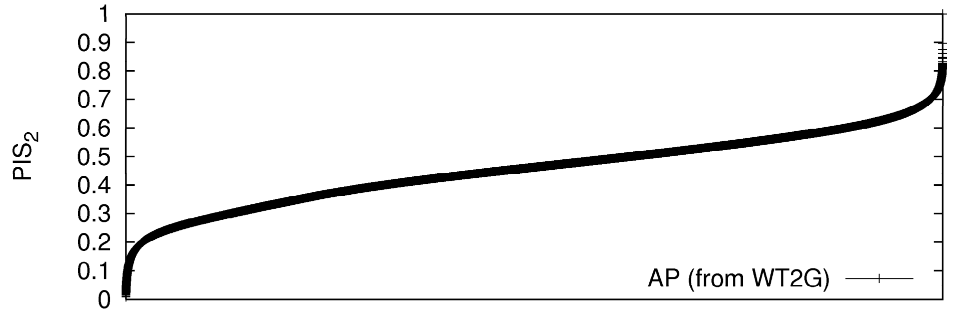


Figure 6.10: PIS_2 of terms in AP (POS 4-grams from WT2G).

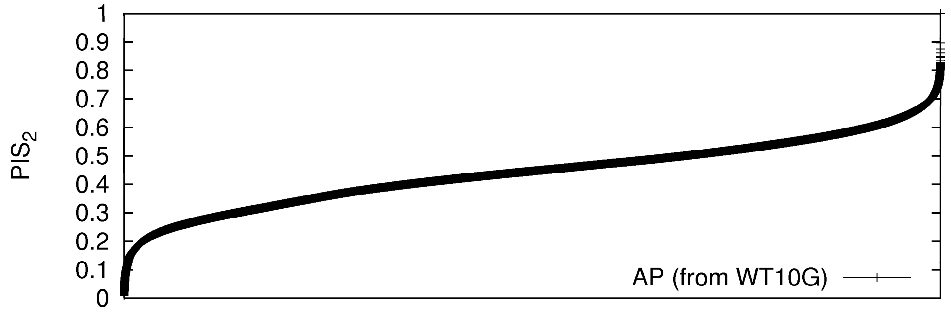


Figure 6.11: PIS_2 of terms in AP (POS 4-grams from WT10G).

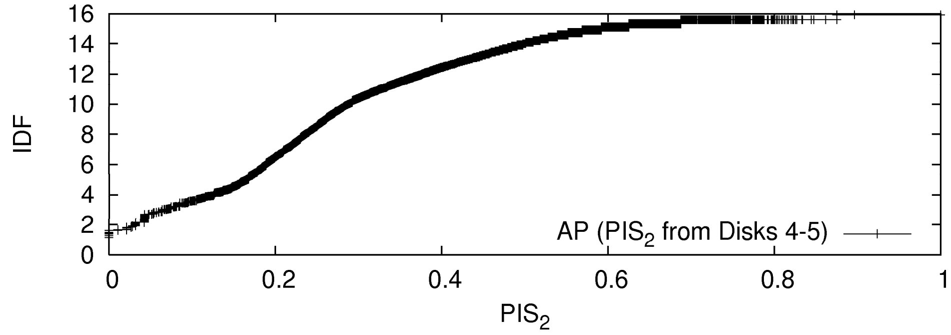


Figure 6.12: PIS_2 versus IDF of terms in AP (POS 4-grams from Disks 4&5).

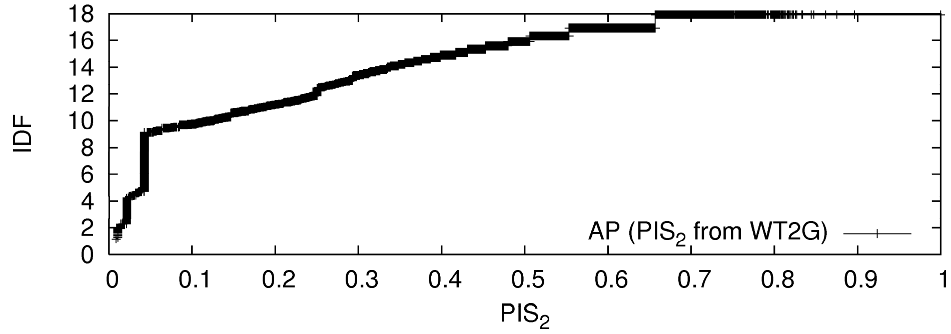


Figure 6.13: PIS_2 versus IDF of terms in AP (POS 4-grams from WT2G).

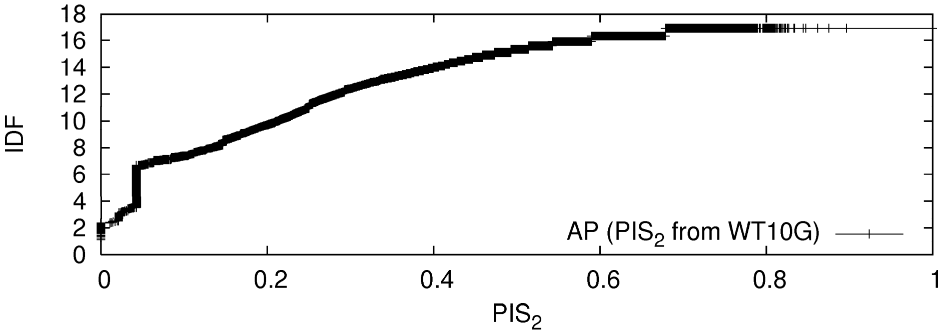


Figure 6.14: PIS$_2$ versus IDF of terms in AP (POS 4-grams from WT10G).

6.6 Summary

This chapter showed that POS n-grams are distributed in language similarly to words, i.e., in a Zipfian distribution. This chapter also showed that the term information score computed from POS n-grams, proposed in Chapter 5, is correlated to IDF. These points were validated in five TREC collections of varying size, word statistics, and domain, and under unbiased experimental settings. The next chapter suggests how POS n-gram frequency and the two PIS alternatives can be used for IR.

Chapter 7

Applications to information retrieval

7.1 Introduction

So far, this thesis has presented empirical evidence suggesting that there exists a relationship between POS n-gram frequency and informative content. Based on this relationship, a term information score called PIS has been proposed, which is derived exclusively from POS n-grams. Specifically, two versions of PIS have been presented, called PIS$_1$ and PIS$_2$. This chapter suggests applications of POS n-gram frequency and of PIS to IR. This chapter is split into two parts:

- Section 7.2 presents and evaluates two applications of POS n-gram frequency to IR which remove content-poor text from queries and documents, namely to reformulate queries (Section 7.2.2) and to prune noise from the index of an IR system (Section 7.2.3).
- Section 7.3 presents and evaluates applications of PIS to IR, namely as an alternative to IDF (Section 7.3.2), and also as additional evidence that can enhance overall retrieval performance (Section 7.3.3).

7.2 Part of speech n-gram frequency

7.2.1 Introduction

This section presents two IR applications that use the frequency of POS n-grams in a collection in order to detect content-poor text. The initial motivation for these applications was the observation that, when ranking all the POS n-grams in a collection according to their frequency, the most frequent POS n-grams had a tendency to contain mostly open class parts of speech, and the least frequent POS n-grams had a tendency to contain mostly closed class parts of speech. This point was presented in Section 5.2. Based on this observation, this section suggests two applications which use POS n-gram frequency to detect and remove content-poor text from either queries (query reformulation) or documents (index pruning).

7.2.2 Part of speech-based query reformulation

7.2.2.1 Experimental hypothesis

Experiments are conducted to test the hypothesis that the frequency of POS n-grams in a collection can be used to detect content-poor text. Removing such content-poor text from verbose queries can render them more informative, hence they can fetch more informative documents. This can benefit retrieval effectiveness.

7.2.2.2 Experimental methodology

The experiments are organised as follows. The setting is a retrieval system, implementing an established retrieval model, and matching documents to queries from standard TREC datasets. The baseline is matching documents to full TREC queries, using two standard probabilistic models at default settings. To test the hypothesis, documents are matched to queries that have been reformulated according to POS n-gram frequency. Specifically, POS n-gram frequency is used to remove content-poor text from queries as follows:

1. Given a POS tagged collection, all POS n-grams are extracted from it and ranked by their frequency in the collection. A threshold θ of POS n-gram frequency is defined, so that POS n-grams below the threshold are assumed to be too infrequent to be informative. These POS n-grams are considered content-poor.

2. The queries of the experiments are POS tagged, and all POS n-grams are extracted from them.

3. Each POS n-gram in the queries is compared to the list of POS n-grams extracted from the collection. For each POS n-gram below the threshold

θ, its corresponding terms are removed from the query, on the assumption that they are content-poor.

7.2.2.3 Experimental settings

As explained in Section 2.7, within TREC-style evaluation, each collection has an associated set of queries and relevance assessments. Here, for retrieval, two standard TREC collections are used: WT2G and WT10G, initially presented in Section 6.3.2, Table 6.3, page 80, and their corresponding queries: queries 401-450 for WT2G, and queries 451-550 for WT10G. Even though any of the five standard TREC datasets presented in Section 6.3.2 could be used, these two datasets are selected because there exists previous literature describing very similar applications, to which the results described here can be compared directly (Section 7.2.2 compares performance to best TREC scores for these datasets). For these experiments very long queries are used, which include all TREC topic fields (`title + description + narrative`). The reason for using very long queries is that, for this application, queries need to be POS tagged, and short queries which contain few keywords cannot be POS tagged accurately.

The pre-processing involved in these retrieval experiments is exactly the same as the pre-processing reported in Section 6.4: in brief, terms are tokenised on whitespace and punctuation marks, and lower-cased; stopwords are removed and terms are stemmed. Two different probabilistic models are used to match documents to queries: BM25 (Robertson & Walker, 1994) and PL2 (Amati, 2003). These models were introduced in Section 2.4.2.3. BM25 and PL2 include parameters, which are set at default settings for these experiments: for BM25 b=0.75 (Robertson & Walker, 1994), and for PL2 c=7 (Amati, 2003, Chapter 7). Tuning these parameters to achieve more competitive retrieval performance is discussed later in Section 7.3.3.4.4.

POS n-grams are extracted from the same collection used for retrieval. The pre-processing involved in POS tagging a collection and extracting POS n-grams from it are exactly as reported in Section 6.4: the collections are POS tagged with the TreeTagger, and POS 4-grams are extracted. These choices of POS tagger and POS n-gram order n were discussed in Section 6.3. Terms which correspond to POS 4-grams of low frequency are removed, where this frequency is bounded by a threshold θ. For these experiments, θ = 20,000, a setting chosen empirically. (The focus of these experiments is to illustrate the proposed use of POS n-grams for IR. This is why parameters are at default or empirically set values, and not

Very long queries			
WT2G			
settings	eval.	Full queries	POS reduced queries
BM25	MAP	0.280	0.298 (+6.4%)
	P10	0.460	**0.480 (+4.3%)**
PL2	MAP	0.265	**0.313 (+18.1%)***
	P10	0.453	0.468 (+3.3%)*
WT10G			
settings	eval.	Full queries	POS reduced queries
BM25	MAP	0.234	0.248 (+6.0%)
	P10	0.394	**0.401 (+1.8%)**
PL2	MAP	0.235	**0.260 (+10.6%)***
	P10	0.383	0.396 (+3.4%)*

Table 7.1: Retrieval performance for very long queries.

MAP in WT2G			
	Full queries		POS reduced queries
settings	long	very long	very long
BM25	0.293	0.280	**0.298**
PL2	**0.319**	0.265	0.313
MAP in WT10G			
	Full queries		POS reduced queries
settings	long	very long	very long
BM25	0.233	0.234	**0.248**
PL2	0.257	0.235	**0.260**

Table 7.2: Retrieval performance for long and very long queries.

optimised for retrieval performance.)

7.2.2.4 Experimental results

This section presents the evaluation results of the retrieval experiments which test whether removing content-poor text from verbose queries, using POS *n*-gram information only, can improve retrieval precision (both MAP and P10).

The experimental results are presented in Table 7.1. Best MAP and P10 scores for each collection are printed in bold. The asterisk * shows statistical significance at p <0.05, according to the Wilcoxon matched-pairs signed-ranks test.

Table 7.1 shows that removing content-poor text from very long queries using

POS *n*-grams produces a higher measure of retrieval performance at all times in all cases, which is statistically significant in some cases. This improvement is consistent for both collections and evaluation measures, and generally considerable: between +3.3% and +18.1%, both with a statistical significance, for WT2G; between +3.4% - +10.6% with a statistical significance, for WT10G. Also, this improvement in retrieval performance over the baseline is always more for MAP than for P10. This indicates that the documents retrieved are overall more relevant, less at the top ranks (precision) and more at the lower ranks (recall).

In order to place the retrieval performance reported in Table 7.1 into context, Table 7.2 compares them to the retrieval performance of shorter queries, under the exact same settings. Table 7.2 shows that removing content-poor text from very long queries using POS *n*-grams improves retrieval performance with respect to different query lengths. This improvement is generally consistent, the only exception being the WT2G collection, for which retrieval with PL2 from long queries slightly outperforms retrieval with POS-based reduced queries (MAP is 0.319 versus 0.313). Note that the MAP scores displayed in Tables 7.1 & 7.2 also compare favourably to the high-scoring equivalent TREC runs, namely, MAP=0.324 (and MAP=0.383 when using Web evidence) Robertson & Walker (1994) for TREC-8, and MAP=0.269 Fujita (2001) for TREC-9.

Overall, the experimental evidence in Tables 7.1 & 7.2 validates the hypothesis presented in Section 7.2.3.1 that the frequency of POS *n*-grams in a collection can be used to detect content-poor text, because it shows that removing such content-poor text from long queries can benefit retrieval performance.

A limitation of the POS-based query reformulation technique presented in this section is that it can be applied to verbose queries only, because it requires that queries are previously POS tagged. A natural extension of this technique is it to apply it to whole documents, not only queries. Such an application is presented next.

7.2.3 Part of speech-based index pruning

7.2.3.1 Experimental hypothesis

Experiments are conducted to test the hypothesis that the frequency of POS *n*-grams in a collection can be used to detect content-poor text. Removing such content-poor text from documents means that less resources are needed to index and store these documents in the system. This can benefit retrieval efficiency. This application is called index pruning, because by removing text from documents, the overall size of the document index is reduced. (The index of IR systems was presented in Section 2.3.2.)

7.2.3.2 Experimental methodology

The baseline of these experiments is the same two probabilistic models for matching documents to queries (BM25 and PL2) used in Section 7.2.2, at default settings. To test the hypothesis, POS n-gram frequency is used to remove content-poor text from documents. The general methodology is the same as the one used for query reformulation in Section 7.2.2:

1. Given a POS tagged collection, all POS n-grams are extracted from it and ranked by their frequency in the collection. A threshold θ of POS n-gram frequency is defined, so that POS n-grams below the threshold are assumed to be too infrequent to be informative. These POS n-grams are considered content-poor.

2. The documents of the experiments are POS tagged, and all POS n-grams are extracted from them.

3. Each POS n-gram in the documents is compared to the list of POS n-grams extracted from the collection. For each POS n-gram below the threshold θ, its corresponding terms are removed from the document, on the assumption that they are content-poor.

7.2.3.3 Experimental settings

The TREC datasets used in these experiments are the same as the ones used in Section 7.2.2: WT2G and WT10G, initially presented in Section 6.3.2, Table 6.3, page 80, and their corresponding queries: queries 401-450 for WT2G, and queries 451-550 for WT10G. Even though any of the five standard TREC datasets presented in Section 6.3.2 could be used, these two datasets are selected because there exists previous literature describing very similar applications, to which the results described here can be compared directly (Section 7.2.3 compares performance to past state of the art applications for these datasets). For these experiments, very short queries are used (`title` only), because they are more realistic of real queries used on the Web (Ozmutlu *et al.*, 2004).

The pre-processing involved in retrieval is exactly as reported in Section 6.4: in brief, terms are tokenised on whitespace and punctuation marks, and lower-cased; stopwords are removed and terms are stemmed. Two different probabilistic

POS 4-grams	% pruned from full index					
	WT2G			WT10G		
	tokens	terms	postings	tokens	terms	postings
23,000	18.39	5.07	14.56	16.57	4.68	14.30
22,000	12.13	3.55	9.46	11.16	3.23	9.38
21,000	8.47	2.16	6.53	7.90	1.96	6.56
20,000	6.00	1.46	4.57	5.70	1.39	4.65
19,000	4.38	1.10	3.31	4.24	1.08	3.41
18,000	3.24	0.82	2.44	3.33	0.84	2.55
17,000	2.44	0.65	1.83	2.43	0.66	1.93
16,000	1.84	0.50	1.37	1.86	0.53	1.46
15,000	1.41	0.39	1.04	1.44	0.40	1.11
14,000	1.10	0.31	0.80	1.13	0.31	0.86
13,000	0.82	0.24	0.61	0.86	0.25	0.66

Table 7.3: Collection statistics after pruning.

models are used to match documents to queries: BM25 (Robertson & Walker, 1994) and PL2 (Amati, 2003). These models were introduced in Section 2.4.2.3. BM25 and PL2 include parameters, as discussed in Section 7.3.3.4.4, page 122. The default values of these parameters are used, namely b=0.75 (Robertson & Walker, 1994), and c=7 (Amati, 2003, Chapter 7), respectively.

POS n-grams are extracted from the same collection used for retrieval. The pre-processing involved in POS tagging a collection and extracting POS n-grams from it are exactly as reported in Section 6.4: collections are POS tagged with the TreeTagger, and POS 4-grams are extracted. Terms which correspond to POS 4-grams of low frequency are removed, where this frequency is bounded by a threshold θ. For these experiments, θ = 20,000, a setting chosen empirically. (The focus of these experiments is to illustrate the proposed use of POS n-grams for IR. This is why parameters are at default or empirically set values, and not optimised for retrieval performance.)

7.2.3.4 Experimental results

This section presents the evaluation results of the retrieval experiments which test whether removing content-poor text from documents, using POS n-gram information only, can improve retrieval efficiency. The general idea of these experiments is to remove content-poor text from documents, so that the overall size of the index is reduced, without significant effects on retrieval effectiveness, since content-poor text is of little use to retrieval.

Table 7.3 presents how much the overall size of the index is reduced, when terms corresponding to low-frequency This is shown for two different collections, and with respect to a reduction measured in (i) tokens, (ii) individual terms, and (iii) postings. (Postings are the document pointers for a term that are contained in the inverted list, as presented in Section 2.3.2.1.) The first column of Table 7.3 corresponds to the value of the threshold θ, i.e. how many low-frequency POS n-grams are used. For example, $\theta = 13{,}000$ means that the terms removed correspond to the 13,000 least frequent POS 4-grams in the collection. Hence, the more θ increases, the more the resulting index pruning.

Table 7.4 presents the retrieval performance corresponding to each level of index pruning shown in Table 7.3, separately for each collection. Pruning levels are reported in % reduction of postings. Best MAP and P10 scores for each collection are printed in bold. Italics denote scores equal to or better than the baseline. The asterisk * (**) shows statistical significance at p <0.05 (p <0.01), according to the Wilcoxon matched-pairs signed-ranks test. Table 7.4 shows that light pruning leads to an overall improvement in MAP and P10 over the full index, which is sometimes statistically significant. Two important observations are drawn from this table. Firstly, at no point does pruning hurt significantly retrieval. This point is very encouraging, considering that the POS n-gram pruning technique uses no document-specific criteria. Secondly, light pruning can improve both MAP and P10. In fact, the best obtained MAP and P10 scores for WT2G, namely MAP = 0.317 and P10 = 0.467, are not given by the full index, but by slightly pruning the index. Both of these scores are statistically very significant (p << 0.01). For WT10G, the best overall MAP score, namely MAP = 0.209, is given by slightly pruning the index, but the best P10 (P10 = 0.324) is given by the full index.

WT2G				
POS -based index	MAP		P10	
compression	BM25	PL2	BM25	PL2
14.56%	0.241 (-6.2%)**	0.297 (-6.1%)**	*0.431 (+1.4%)***	0.453 (-0.4%)**
9.46%	0.250 (-2.4%)	0.301 (-4.7%)	*0.429 (+0.9%)*	*0.455 (none)*
6.53%	0.251 (-2.0%)	0.303 (-4.0%)	*0.429 (+0.9%)*	*0.464 (+2.0%)*
4.57%	*0.255 (-0.4%)*	0.309 (-1.9%)	**0.441 (+3.8%)**	*0.461 (+1.3%)*
3.31%	*0.257 (+0.4%)*	0.310 (-1.6%)	**0.441 (+3.8%)**	*0.460 (+1.1%)*
2.44%	*0.257 (+0.4%)*	0.310 (-1.6%)	**0.441 (+3.3%)**	*0.461 (+1.3%)*
1.83%	*0.259 (+1.2%)*	0.314 (-0.3%)	**0.441 (+3.8%)**	*0.463 (+1.8%)*
1.37%	**0.260 (+1.6%)***	*0.316 (+0.3)***	*0.436 (+2.6%)**	**0.467 (+2.6%)****
1.04%	**0.260 (+1.6%)***	*0.316 (+0.3%)**	*0.436 (+2.6%)**	*0.458 (+0.6%)**
0.80%	*0.259 (+1.2%)*	**0.317 (+0.6%)**	*0.433 (+1.9%)*	*0.459 (+0.9%)*
0.61%	**0.260 (+1.6%)***	**0.317 (+0.6%)***	*0.434 (+2.1%)**	*0.459 (+0.9%)**
full index	0.256	0.315	0.425	0.455

WT10G				
POS - based index	MAP		P10	
compression	BM25	PL2	BM25	PL2
14.30%	0.175 (-6.9%)**	0.196 (-6.7%)**	0.293 (-2.4%)**	0.297 (-9.1%)**
9.38%	0.182 (-2.7%)*	0.204 (-2.0%)	**0.304 (+1.3%)***	0.306 (-5.9%)
6.56%	0.185 (-1.1%)*	0.205 (-1.5%)**	*0.302 (+0.7%)**	0.315 (-2.8%)**
4.65%	*0.187 (none)*	0.206 (-1.0%)	*0.300 (none)*	0.316 (-2.5%)
3.41%	0.186 (-0.5%)	0.205 (-1.5%)	*0.301 (+0.3%)*	0.311 (-4.2%)
2.55%	*0.187 (none)*	0.207 (-0.5%)*	0.298 (-0.7%)	0.318 (-1.9%)*
1.93%	*0.187 (none)*	*0.208 (none)*	*0.300 (none)*	0.322 (-0.6%)
1.46%	0.186 (-0.5%)	*0.208 (none)*	*0.302 (+0.7%)*	0.323 (-0.6%)
1.11%	*0.187 (none)*	*0.208 (none)*	*0.302 (+0.7%)*	0.323 (-0.6%)
0.86%	*0.187 (none)*	*0.208 (none)*	*0.302 (+0.7%)*	**0.324 (none)**
0.66%	**0.188 (+0.5%)**	**0.209 (+0.5%)**	*0.303 (+1.0%)*	**0.324 (none)**
full index	0.187	0.208	0.300	**0.324**

Table 7.4: Retrieval performance at different index pruning levels.

Overall, Table 7.4 shows that removing content-poor text from documents using POS n-grams improves retrieval efficiency without significantly harming retrieval effectiveness, at all times. This observation is consistent for both collections and evaluation measures:

- for WT2G, index pruning ranges between -0.61% and -14.56% of the original index size, which is a small gain in retrieval efficiency, with no considerable alternation to retrieval effectiveness (between +1.6% and -6.2% for MAP; between +3.8% and -0.4% for P10);

- for WT10G, index pruning ranges between -0.66% and -14.30% of the original index size, which is a small gain in retrieval efficiency, with no considerable alternation to retrieval effectiveness (between +0.5% and -6.9% for MAP; between +1.3% and -9.1% for P10);

This indicates that the documents retrieved are overall smaller and roughly as relevant as their corresponding full documents. The highest rates of index pruning generally correspond to the highest gain in retrieval efficiency, but also to the highest deterioration in retrieval effectiveness. Balancing these two is central in IR research.

In order to place the figures presented in Table 7.4 into context, they are compared to corresponding figures reported for the same datasets in literature: Table 7.5 compares the POS-based index pruning technique reported in Table 7.4 to other related work of similar index pruning levels. Table 7.5 shows that the POS-based technique is at least comparable to the technique of Carmel *et al.* (2001a,b), which is among the best performing index pruning techniques reported for these datasets. In Table 7.5, † refers to Carmel *et al.* (2001a), and ‡ refers to Carmel *et al.* (2001b). The technique of Carmel *et al.* (2001a,b) removes individual terms from the index, according to their lexical statistics (term frequency, and resulting term weights). This type of evidence is document-based, i.e., a term occurring in document A and document B can be removed from document A, but not from document B, if it is found to contribute to the content of document B but not to the content of document A. On the contrary, the POS-based pruning technique removes term n-grams (not individual terms), according to POS evidence only, which is derived from the whole collection. This type of evidence is not document-specific, i.e., a term n-gram estimated to be content-poor is removed from all documents in the collection.

pruning strategy	index compression	diff. from full index		collection
		MAP	P10	
POS n-grams	14.5%	-6.2%	+1.4%	WT2G
terms†	13.2%	-4.0%	+2.5%	
POS n-grams	9.46%	-2.4%	+0.9%	
POS n-grams	14.3%	-6.9%	-2.4%	WT10G
terms‡	10.7%	-1.9%	none	
POS n-grams	9.38%	-2.7%	+1.3%	

Table 7.5: Comparison of POS n-gram based versus term based pruning.

7.2.4 Conclusion

Section 7.2 tested the hypothesis that POS n-gram frequency in the collection can be used to detect content-poor text from verbose queries and documents, which, if removed, can improve retrieval performance. A series of experiments were conducted, testing this hypothesis first with very long queries (query reformulation application), and then with documents (index pruning application). Experimental evidence using standard models and datasets validated the hypothesis.

7.3 Part of speech n-gram information score

7.3.1 Introduction

Section 5.3 presented a term information score called PIS, which is derived exclusively from POS n-grams. Specifically, two versions of PIS were presented, called PIS_1 and PIS_2. This section suggests applications of PIS_1 and PIS_2 to IR. Firstly, PIS_2 is presented as an alternative to IDF (Section 7.3.2), and secondly PIS_1 is presented as an additional type of evidence that can be used to improve overall retrieval performance (Section 7.3.3).

7.3.2 Alternative to inverse document frequency

Section 6.5 showed that PIS_2 has a strong positive correlation to IDF. The aim of this section is to test whether PIS_2 and IDF are equivalent when used to match documents to queries in an IR system.

The remainder of this section is organised as follows. Section 7.3.2.1 states the experimental hypothesis. Section 7.3.2.2 presents the experimental methodology. Section 7.3.2.3 presents the experimental settings. Section 7.3.2.4 reports and discusses the experimental results, and Section 7.3.2.5 summarises and concludes this section.

7.3.2.1 Experimental hypothesis

The experimental hypothesis is that PIS$_2$ can replace conventional IDF, when matching documents to queries, without causing any significant change in the retrieval performance. The reasoning behind this is that IDF - PIS$_2$ are very strongly correlated, because they both include a term frequency component implicitly in their respective computations. This point was discussed in Section 5.3.4.4.

7.3.2.2 Experimental methodology

The experiments are organised as follows. The setting is a retrieval system, implementing an established model for matching documents to queries from standard TREC datasets. The baseline is a basic TF:IDF vector space model. Two rounds of experiments are conducted:

- **First experiment:** Replace the IDF component of the baseline with PIS$_2$, and compare retrieval performance to that of the baseline:

  $$TF : IDF \Rightarrow TF : PIS_2 \tag{7.1}$$

  TF:PIS2 differs from TF:IDF in only one respect: it replaces IDF, which is computed from term frequencies, with an approximation of IDF, which is computed from POS *n*-grams.

- **Second experiment:** Combine the IDF of the baseline model with PIS$_2$, and compare retrieval performance to that of the baseline:

  $$TF : IDF \Rightarrow TF : IDF : PIS_2 \tag{7.2}$$

  TF:IDF:PIS2 differs from TF:IDF in only one respect: in addition to the baseline components, which are computed from term frequencies, it contains an approximation of IDF, which is computed from POS *n*-grams.

  It is expected that TF:PIS2 and TF:IDF:PIS$_2$ will give similar retrieval perfomance to TF:IDF.

Query	WT10G	Disks 4&5
Title	2.42	2.62
Description	5.28	6.99

Table 7.6: Average query length (in words).

7.3.2.3 Experimental settings

This section presents the experimental settings used in these experiments, separately for retrieval and to compute PIS$_2$. Section 7.3.2.3.1 presents the datasets. Section 7.3.2.3.2 presents the pre-processing involved in the retrieval process and also in computing PIS$_2$. Section 7.3.2.3.3 presents the processing involved in matching documents to queries. Section 7.3.2.3.4 presents the measures used to evaluate retrieval performance.

7.3.2.3.1 Datasets For retrieval, two standard TREC collections are used: WT10G and Disks 4&5. These collections are used because there are more ad-hoc queries available for them, than for the other TREC collections presented in Section 6.3.2. The collection characteristics are displayed in Table 6.3, page 80. Queries 451-550 are used for WT10G, and queries 301-450 and 601-700 are used for Disks 4&5. Two types of queries are used: short (`title`) and long (`description`). These two query types are standard in TREC (Voorhees & Harman, 2001). The average length of these queries is presented in Table 7.6.

PIS$_2$ is computed from the same collection used for retrieval. (Section 6.5 showed that the computation of PIS$_2$ is not collection-dependent.)

7.3.2.3.2 Pre-processing The pre-processing involved in retrieval is exactly as reported in Section 6.4: terms are tokenised on whitespace and punctuation marks, and lower-cased. Stopwords are removed, and words are stemmed with the Porter stemming algorithm (Porter, 1980). The process described above is done using the Terrier IR platform (Ounis *et al.*, 2007).

The pre-processing involved in computing PIS$_2$ is exactly as reported in Section 6.4, page 89: the collections are POS tagged with the TreeTagger, and POS 4-grams are extracted. These choices of POS tagger and POS n-gram order n were discussed in Section 6.3.

Experimental settings overview	
Retrieval collection	-Disks 4&5 -WT10G
POS n-gram collection	-Disks 4&5 -WT10G
Query length	-short (`Title`) -long (`Description`)
Retrieval model	TF:IDF (Eq. 2.5, 2.6, 2.9, pages 22 - 23)
Use of PIS$_2$	-TF:IDF, no PIS$_2$ (baseline) -TF:PIS$_2$, no IDF (PIS$_2$ replaces IDF) (Eq. 7.1, page 110) -TF:IDF:PIS$_2$ (PIS$_2$ and IDF combined) (Eq. 7.2, page 110)

Table 7.7: Settings of the experiments reported in Section 7.3.2.4.

7.3.2.3.3 Processing The processing involved in retrieval is the following: a basic TF:IDF vector space formulation is used to match documents to queries. (Matching models were introduced in Section 2.4.2, and vector space models in particular were presented in Section 2.4.2.2). Specifically, Equation 2.9 is used, which computes the score of a document for a query as the Euclidean distance between a TF and IDF component, for each term t in query q:

$$S_{q,d} = \sum_{t \in q} TF \cdot IDF \approx \sum_{t \in q} w_{t,d} \cdot w_{t,q}$$

where

- $w_{t,d}$ is the weight of a term in the document, given by Equation 2.6, page 22; and
- $w_{t,q}$ is the weight of a term in the query, given by Equation 2.5, page 22.

The processing involved in computing PIS$_2$ is as follows: PIS$_2$ is computed with Equation 5.21, page 73. Equation 5.21 includes the two variables λ and ϱ, which represent the probability that a first and second degree part of speech is informative, respectively (presented in Section 5.3.2). For these experiments, the values of λ and ϱ are derived using Bayes Rule, as shown in Section 5.3.2.2, by setting $\lambda = 1$ and solving for ϱ. The values of λ and ϱ used here are derived from Bayes Rule instead of being tuned to optimise retrieval performance, because the aim of these experiments is to compare PIS$_2$ to IDF on a basic setting, and not to achieve competitive retrieval performance. In Section 7.3.3, where experiments

Short queries				
Disks 4&5				
row	eval.	TF:IDF	TF:PIS$_2$	TF:IDF:PIS$_2$
1	MAP	0.203	**0.204 (+0.5%)***	**0.204 (+0.5%)****
2	P10	0.391	**0.397 (+1.5%)***	**0.397 (+1.5%)****
WT10G				
row	eval.	TF:IDF	TF:PIS$_2$	TF:IDF:PIS$_2$
3	MAP	**0.187**	**0.187 (none)***	**0.187 (none)****
4	P10	**0.297**	**0.297 (none)***	**0.297 (none)****

Table 7.8: Retrieval performance with TF:IDF for short queries (Disks 4&5 and WT10G)

with PIS aim to enhance retrieval performance, λ and ϱ is tuned to optimise retrieval performance.

7.3.2.3.4 Evaluation PIS$_2$ is compared to IDF with respect to retrieval performance. Retrieval performance is evaluated in terms of MAP and P10 (presented in Section 2.7), which are standard measures are standard in the TREC paradigm (Voorhees & Harman, 2001). In order to establish whether the experimental results occurred by chance or not, results of statistical significance tests, using the Wilcoxon matched-pairs signed-ranks test, are reported ($p < 0.05$ is statistically significant, and $p < 0.01$ is statistically very significant).

Table 7.7 summarises the experimental settings.

7.3.2.4 Experimental results

This section presents the experimental results, in Tables 7.8 and 7.9, for short and long queries, respectively. Best score(s) are printed in bold. The asterisk * (**) shows statistical significance at p <0.05 (p <0.01).

Tables 7.8 and 7.9 show that:

1. There is no consistently significant difference in retrieval performance between TF:PIS$_2$ and TF:IDF:PIS$_2$, nor with respect to different query lengths, collections, or evaluation measures.

2. The best overall performance is always associated with PIS$_2$.

The first observation from Tables 7.8 and 7.9 is that there is no consistently significant difference in retrieval performance between either combination of PIS$_2$

Long queries				
Disks 4&5				
row	eval.	TF:IDF	TF:PIS$_2$	TF:IDF:PIS$_2$
1	MAP	0.216	0.217 (+0.5%)*	**0.220 (+1.9%)****
2	P10	0.409	**0.410 (+0.2%)***	**0.410 (+0.2%)****
WT10G				
row	eval.	TF:IDF	TF:PIS$_2$	TF:IDF:PIS$_2$
3	MAP	0.200	0.200 (none)**	**0.203 (+1.5%)****
4	P10	0.324	0.324 (none)**	**0.325 (+0.3%)****

Table 7.9: Retrieval performance with TF:IDF for long queries (Disks 4&5 and WT10G).

into the baseline, nor with regard to query length, collection, or evaluation measure. Specifically, the % difference from the baseline tends to be overall similar,

- for different PIS$_2$ combinations:
  - **TF:PIS$_2$**: between none (Table 7.8, rows 3&4, column 4, and Table 7.9, rows 3&4, column 4) - +1.5% (Table 7.8, row 2, column 4);
  - **TF:IDF:PIS$_2$**: between none (Table 7.8, rows 3&4, last column) - +1.9% (Table 7.9, row 1, last column);
- for different query lengths:
  - **short queries**: between none (Table 7.8, all WT10G) - +1.5% (Table 7.8, row 2, columns 3&4);
  - **long queries**: between none (Table 7.9, rows 3&4, column 4) - +1.9% (Table 7.9, row 3, last column);
- for different collections:
  - **Disks 4&5**: between +0.2% (Table 7.9, rows 1&2, columns 4 & 5) - +1.9% (Table 7.9, row 1, last column);
  - **WT10G**: between none (Table 7.8, all WT10G, and Table 7.9, rows 3&4, column 4) - +1.5% (Table 7.9, row 3, last column);
- for different evaluation measures:
  - **MAP**: between none (Table 7.8, row 3, columns 4&5, and Table 7.9, row 3, column 4) - +1.9% (Table 7.9, row 1, last column);

  - **P10**: between none (Table 7.8, row 4, columns 4&5, and Table 7.9, row 4, column 4) - +1.5% (Table 7.8, row 2, columns 4&5).

This lack of consistently significant difference in retrieval performance validates the hypothesis that PIS_2 can replace IDF without significantly altering retrieval performance.

The second observation from Tables 7.8 & 7.9 is that the best overall performance (printed in bold in the tables) is always associated with PIS_2. The improvement marked when using PIS_2 tends to be overall small, as discussed above, however it is statistically significant at all times. Using PIS_2 in the place of IDF, compared to using both PIS_2 and IDF together, does not alter retrieval performance significantly. This indicates the validity of the hypothesis that PIS_2, computed from POS *n*-grams, can replace IDF, computed from frequency statistics, without significanlty altering retrieval performance.

7.3.2.5 Conclusion

Section 7.3.2 compared a term information score computed from POS *n*-grams (PIS_2) with an established term information score computed from lexical statistics (IDF). Specifically, it tested whether PIS_2 and IDF are equivalent, when used to match documents to queries in an IR system. Experiments with a standard baseline system on two TREC collections showed that PIS_2 can replace IDF without altering significantly retrieval performance, and even improving retrieval performance (not considerably). This conclusion agrees with the observations drawn from Section 6.5, namely that PIS_2 and IDF are very strongly correlated, a fact due to an extent to their common use of term frequency implicitly in their respective computations.

7.3.3 Enhancement to retrieval performance

7.3.3.1 Introduction

The previous section tested whether PIS_2 can replace IDF in an IR system. This section aims to test whether PIS_1 can enhance the retrieval performance of an IR system.

The remainder of this section is organised as follows. Section 7.3.3.2 states the experimental hypothesis. Section 7.3.3.3 presents the experimental methodology. Section 7.3.3.4 presents the experimental settings. Section 7.3.3.5 reports and discusses the experimental results, and Section 7.3.3.6 concludes Section 7.3.

7.3.3.2 Experimental hypothesis

The hypothesis is that using PIS_1 when retrieving documents with respect to queries can improve retrieval performance. The reasoning behind this is that, when computing how informative document terms are with respect to query terms, non-topical information (given by PIS_1) can be used to 'boost' the score of generally informative terms, and similarly decrease the score of generally non-informative terms.

7.3.3.3 Experimental methodology

The experiments are organised as follows. Similarly to Section 7.3.2.2, the setting is a retrieval system, implementing an established retrieval model, and matching documents to queries from standard TREC datasets. The baseline is a standard competitive probabilistic model for matching documents to queries. The hypothesis is tested by integrating PIS_1 into the matching model, and comparing its retrieval performance to that of the baseline. Two rounds of experiments are conducted to test the hypothesis, i.e., PIS_1 is integrated into the matching model in two different ways:

- **First experiment:** PIS_1 is combined with the term frequency component of the retrieval model ($f_{t,d}$). Specificaly, $f_{t,d}$ is the frequency of a term in a document. With this integration, the **input of the matching process** is altered, which computes how important a term is to a document.

- **Second experiment:** PIS_1 is combined with the final weight of a term in a document with respect to a query ($w_{t,d}$). Hence, the **output of the matching process** is altered.

These integrations are discussed separately in Sections 7.3.3.3.1 and 7.3.3.3.2.

Separately for each integration, two further rounds of experiments are realised:

- with default retrieval settings; and

- with retrieval settings optimised for retrieval performance.

The aim is to show that PIS_1 can improve retrieval performance on any setting, i.e., in a robust way. The default and optimal settings are presented in Section 7.3.3.4.4.

The next section discusses the integration of PIS_1 into the retrieval model.

7.3.3.3.1 First integration The previous section suggested two ways of integrating PIS$_1$ into the retrieval model: one that alters the input of the matching process, and one that alters the output of the matching process. For the first integration, PIS$_1$ is multiplied to the frequency of a term in a document, $f_{t,d}$:

$$f_{t,d} \Rightarrow f_{t,d} \cdot PIS_1 \tag{7.3}$$

Equation 7.3 states that $f_{t,d}$ is replaced by $f_{t,d}\cdot$ PIS$_1$ in the retrieval model. This takes place at the beginning, before the model processes $f_{t,d}$. Hence, PIS$_1$ is part of the **input of the matching process**.

PIS$_1$ is combined to $f_{t,d}$ for the following reason: PIS$_1$ is the probability of a term being informative in a non-topical way (Section 5.3.4). As such, it characterises terms. $f_{t,d}$ also characterises terms, but in a topical way (with respect to the topic of a document). It seems appropriate to combine PIS$_1$ to $f_{t,d}$, in the sense that they should complement each other. Specifically, their combination should be an estimation of how informative a term is in general and also in a document.

Different matching models process $f_{t,d}$ differently. For example, often $f_{t,d}$ is normalised according to document length, as presented in Section 2.3.2.2. This means that, when PIS$_1$ is integrated into the matching process as shown in Equation 7.3, PIS$_1$ will also be normalised according to document length. For $f_{t,d}$, normalisation according to document length makes sense, as discussed in Section 2.3.2.2. However, for PIS$_1$, document length normalisation is not intuitive, because of two reasons:

- PIS$_1$ is an approximation of a probability, derived from a computation that already includes a form of normalisation: in Equation 5.14, page 69, the probability that a term is informative, computed from POS n-grams, is 'normalised' by the number of all POS n-grams in the collection. On the contrary, $f_{t,d}$ is a 'raw' frequency count, which has not been normalised. In this respect, PIS$_1$ and $f_{t,d}$ differ, not only in what they represent (non-topical term content - topical term content), but also in their 'statistical properties' (approximated probability - frequency count).

- PIS$_1$ represents the non-topical informative content of a term, whereas $f_{t,d}$ represents the topical informative content of a term. Topical information is by definition related to the topic of the document in which a term occurs. Hence, topical information can be affected by document length. On the contrary, non-topical term content is not related to the document in which

> a term occurs. (PIS$_1$ is computed from statistics derived from a whole collection, and uses no document-specific information.) Hence, PIS$_1$ should not be affected by document length.

The next section presents the second integration of PIS$_1$ into the retrieval process.

7.3.3.3.2 Second integration The second integration combines PIS$_1$ with the weight of a term in a document $w_{t,d}$, i.e., to the **output of the matching process**. Contrary to the first integration, PIS$_1$ undergoes no processes, such as document length normalisation; it is simply multiplied to $w_{t,d}$:

$$w_{t,d} \Rightarrow w_{t,d} \cdot PIS_1 \tag{7.4}$$

Equation 7.4 states that $w_{t,d}$ is replaced by $w_{t,d} \cdot PIS_1$ in the model that matches documents to queries. This type of integration resembles the way prior probabilities, or *priors* are sometimes integrated into the retrieval process (Craswell *et al.*, 2005; Kraaij *et al.*, 2002). Even though their integration can be similar, there is a fundamental difference between PIS$_1$ and such priors: Typically, these priors represent the likelihood of a document being relevant, i.e., they apply to documents, not individual terms. On the contrary, PIS$_1$ represents the likelihood of a term being informative, i.e., it applies to individual terms.

In both integrations presented in Sections 7.3.3.3.1 and 7.3.3.3.2, the combination of PIS$_1$ is realised by multiplication. These integrations are neither exclusive, nor necessarily optimal. They are suggested to illustrate the usability of PIS$_1$ to IR. Multiplication is chosen because PIS$_1$ is an approximated probability, as presented in Chapter 5. As such, its range is 0 - 1. The ranges of $f_{t,d}$ and $w_{t,d}$ are far greater than 1. This difference in magnitude means that simple summation would be unfair and ineffective. Summation of logs could be an alternative. However, using $\log f_{d,t}$ and $\log w_{t,d}$ instead of $f_{t,d}$ and $w_{t,d}$ would mean tampering with the computations of the matching models. In any case, probability multiplication and summation of log probability are generally considered to be approximately equivalent.

The next section presents the experimental settings.

7.3.3.4 Experimental settings

This section presents the experimental settings used in these experiments, separately for retrieval, and to compute PIS$_1$.

Section 7.3.3.4.1 presents the datasets used. Section 7.3.3.4.2 presents the pre-processing involved in the retrieval process and also in computing PIS_1 from POS n-grams. Section 7.3.3.4.3 presents the processing involved in matching documents to queries. Section 7.3.3.4.4 presents how parameters or variables involved in the experiments are tuned. Section 7.3.3.4.5 presents the evaluation measures used to evaluate retrieval performance.

7.3.3.4.1 Datasets The TREC datasets (test collections, topics, and relevance judgements) used in these experiments are the same as the ones presented in Section 7.3.2.3.1. The choice of these datasets was justified in page 111. In brief, the WT10G and Disks 4&5 collections are used, initially presented in Section 6.3.2, Table 6.3, page 80, and their corresponding short and long queries: queries 451-550 for WT10G, and queries 301-450 & 601-700 for Disks 4&5. The average length of these queries was presented in Table 7.6, page 111.

Similarly to the computation of PIS_2 reported in Section 7.3.2.3.1, PIS_1 is computed from the same collection used for retrieval. (Section 6.5 showed that the computation of PIS_1 is collection-independent.)

7.3.3.4.2 Pre-processing The pre-processing involved in retrieval is exactly as reported in Section 7.3.2.3.3: in brief, terms are tokenised on whitespace and punctuation marks, and lower-cased; stopwords are removed and terms are stemmed.

Similarly, the pre-processing involved in computing PIS_1 is exactly as reported in Section 6.3: collections are POS tagged with the TreeTagger, and POS 4-grams are extracted.

7.3.3.4.3 Processing For retrieval, two different probabilistic models are used to match documents to queries. (Retrieval models were introduced in Section 2.4.2, and probabilistic models in particular were presented in Section 2.4.2.3.) The two models are: Okapi's Best Match 25 (BM25) (Robertson & Walker, 1994) and Poisson Laplace 2 (PL2) from the Divergence From Randomness (DFR) framework (Amati, 2003). The formulae of these models are repeated next, in order to point out how PIS_1 is integrated into them.

Regarding BM25, Equations 2.11 & 2.15, pages 24 & 25, state that BM25 computes the relevance score of a document d for a query q as follows:

$$S_{q,d} = \sum_{t \in q} w_{t,q} \cdot w_{t,d}$$

$$= \sum_{t \in q} \log(\frac{N - f_t + 0.5}{f_t + 0.5}) \cdot \frac{(k_3 + 1) \cdot f_{t,q}}{k_3 + f_{t,q}} \cdot \frac{(k_1 + 1) \cdot f_{t,d}}{K + f_{t,d}}$$

where

- N is the number of documents in the collection;
- f_t is the frequency of documents containing term t in the collection;
- k_3 is a parameter, the recommended value of which is 1000 (Robertson & Walker, 1994);
- $f_{t,q}$ is the term frequency in the query;
- k_1 is a parameter, the recommended value of which is 1.2 (Robertson & Walker, 1994);
- $f_{t,d}$ is the term frequency in the document; and
- K is given by:

  $$K = k_1((1 - b) + b \cdot \frac{dl}{avdl})$$

  where

  - b is a parameter, the recommended value of which is 0.75 (Robertson & Walker, 1994);
  - dl is the document length, measured in any suitable units (e.g. indexed terms, bytes, and so on); and
  - $avdl$ is the average document length in the collection, measured similarly to dl.

For the first integration (Section 7.3.3.3.1, Equation 7.3, page 117), PIS$_1$ is multiplied to $f_{t,d}$, which affects the model as follows:

$$\frac{(k_1 + 1) \cdot f_{t,d}}{K + f_{t,d}} \Rightarrow \frac{(k_1 + 1) \cdot f_{t,d} \cdot PIS_1}{K + (f_{t,d} \cdot PIS_1)} \tag{7.5}$$

For the second integration (Section 7.3.3.3.2, Equation 7.4, page 118), PIS$_1$ is multiplied to $S_{q,d}$, which affects the model as follows (using the concise formulation):

$$S_{q,d} = \sum_{t \in q} w_{t,q} \cdot w_{t,d} \cdot PIS_1 \tag{7.6}$$

Regarding PL2, Equations 2.16 & 2.23, pages 26 & 27, state that PL2 computes the relevance score of a document d for a query q as follows:

$$S_{q,d} = \sum_{t \in q} w_{t,q} \cdot w_{t,d}$$

$$= \sum_{t \in q} w_{t,q} \cdot \frac{1}{tfn + 1}(tfn \cdot \log_2 \frac{tfn}{\lambda} \cdot \log_2 e + 0.5 \cdot \log_2 (2\pi \cdot tfn))$$

where tfn is the normalised frequency of a term in a document, given by Equation 2.22, page 27 as follows:

$$tfn = f_{t,d} \cdot \log_2 (1 + c \cdot \frac{avdl}{dl})$$

where

- c is a parameter, the recommended value of which is 7.0 (Amati, 2003);
- dl is the document length, measured in any suitable units (e.g. indexed terms, bytes, and so on); and
- $avdl$ is the average document length in the collection, measured similarly to dl.

For the first integration (Section 7.3.3.3.1, Equation 7.3, page 117), PIS$_1$ is multiplied to $f_{t,d}$. This affects the tfn component (Equation 2.22) of the model as follows:

$$f_{t,d} \cdot \log_2 (1 + c \cdot \frac{avdl}{dl}) \Rightarrow (f_{t,d} \cdot PIS_1) \cdot \log_2 (1 + c \cdot \frac{avdl}{dl}) \qquad (7.7)$$

For the second integration (Section 7.3.3.3.2, Equation 7.4, page 118), PIS$_1$ is multiplied to $S_{q,d}$. This affects the model as follows (using the concise PL2 formulation of Equation 2.16):

$$S_{q,d} = \sum_{t \in q} w_{t,q} \cdot w_{t,d} \cdot PIS_1 \qquad (7.8)$$

Both BM25 and PL2 include parameters. The next section presents how these parameters, and also any other parameters or variables involved in the experiments are set.

7.3.3.4.4 Default and optimal settings As mentioned in Section 7.3.3.3, all experiments are conducted twice: (i) with default settings, and (ii) with settings optimised for retrieval precision. The aim is to evaluate the hypothesis first on standard settings, and then on a stronger baseline. Settings here means parameters or variables in the retrieval models and in the computation of PIS_1. Specifically, these are:

- parameter b of the retrieval model BM25 (Equation 2.11, page 24);
- parameter c of the retrieval model PL2 (Equation 2.16, page 26); and
- variables λ and ϱ[1] of the computation of PIS_1 (Equation 5.14, page 69).

BM25 includes further parameters (k_1 and k_3); these do not have a significant impact on retrieval performance (Robertson & Walker, 1994), hence their recommended values are used and they are treated as constants. These values are k_1= 1.2 and k_3= 1000 (Robertson & Walker, 1994).

Default parameters For retrieval, parameters b of BM25 and c of PL2 have a ‘smoothing’ role, i.e., they normalise the relevance score of a document for a query across document lengths, in order to avoid bias towards longer documents. The default/recommended settings are b=0.75 (Robertson & Walker, 1994), and c=7 (Amati, 2003, Chapter 7). For computing PIS_1, the values of λ and ϱ are derived using Bayes Rule, as shown in Section 5.3.2.2, by setting $\lambda = 1$ and solving for ϱ.

Parameters optimised for retrieval The setting of b and c depends on the collection and the query set, and has been shown to have an important impact on retrieval performance (Chowdhury *et al.*, 2002; He & Ounis, 2003, 2005b). Generally, the less sensitive a retrieval model is to changes in its parameters, the more robust it is. Hence, by comparing system performance on default and optimal settings, we can evaluate how robust the system is.

Parameters b, c, λ, and ϱ are optimised for retrieval performance, and specifically MAP and P10 separately, by training using data sweeping over a large range of values.

The range of the parameter values are:

[1] λ and ϱ are probability approximations, not parameters. Here, they are treated as variables, and tuned to optimise retrieval performance.

- for b: within (0,1] with a unique interval of 0.05.
- for c: within [1,32] with an increasing interval:
  - from 1 to 4 with an interval of 1,
  - from 6 to 8 with an interval of 2,
  - from 12 to 16 with an interval of 4,
  - from 24 to 32 with an interval of 8.

  The ten values sampled are 1, 2, 3, 4, 6, 8, 12, 16, 24, and 32.
- for λ and ϱ: within (0,1] with a unique interval of 0.05.

For BM25 and PL2, the chosen ranges are believed to be wide enough to cover the optimal settings of BM25 and PL2, according to He & Ounis (2003, 2005a,b). For λ and ϱ, this range follows from the fact that λ and ϱ are approximated probabilities. For each parameter, the value selected gives the best MAP/P10 performance.

This optimisation is repeated separately for short and long queries, separately for MAP and P10, and also separately for:

- the baseline retrieval model (no PIS$_1$, hence no λ and ϱ); and
- the retrieval model with PIS$_1$.

In the second case, when using the retrieval model with PIS$_1$, there are three parameters: (i) b or c depending on the model, (ii) λ, and (iii) ϱ. All three parameters are optimised over all possible combinations.

All parameter values (default and optimised) are shown in Tables E.1 and E.2, Appendix E, pages 176 - 177.

7.3.3.4.5 Evaluation Retrieval performance is evaluated in terms of MAP and P10, and the results of statistical significance tests, using the Wilcoxon matched-pairs signed-ranks test are reported.

An additional evaluation measure is used to evaluate how robust retrieval performance is. Specifically, Section 7.3.3.4.4 stated that comparing retrieval performance across a range of parameter values reveals whether changes are due to PIS$_1$ or to the model parameters. The effect of this parameter tuning upon the model robustness can be quantitatively analysed using the parameter sensitivity

Experimental settings overview	
Retrieval collection	-Disks 4&5 -WT10G
POS n-gram collection	-Disks 4&5 -WT10G
Query length	-short (`title`) -long (`description`)
Retrieval model	-BM25 (Equation 2.11, page 24) -PL2 (Equation 2.16, page 26)
Use of PIS$_1$	-no PIS$_1$ (baseline) -PIS$_1$ $\times f_{t,d}$ (input of matching process) -PIS$_1$ $\times w_{t,d}$ (output of matching)
Settings	-default -optimised for MAP -optimised for P10

Table 7.10: Settings for the experiments reported in Section 7.3.3.5.

Spread measure, which measures the flatness of a posterior distribution over a set of parameter values (Metzler, 2006):

$$S = m(max, X) - m(min, X) \tag{7.9}$$

By substituting $m(max, X)$ (resp. $m(min, X)$) for maximum MAP/P10 (resp. minimum MAP/P10) in Equation 7.9, one can observe the flatness of the MAP/P10 distribution across a range of parameter values. This indicates how sensitive the retrieval model is to parameter tuning: smaller spread S indicates less sensitivity to parameter tuning, hence more robust models.

The experimental settings presented in Section 7.3.3.4 are summarised in Table 7.10.

7.3.3.5 Experimental results

This section presents the evaluation results of the retrieval experiments. The aim of these experiments is to test whether using PIS$_1$ for retrieval is beneficial. Section 7.3.3.5.1 presents results on retrieval precision (both MAP and P10). Section 7.3.3.5.2 presents results on retrieval robustness.

7.3.3.5.1 Retrieval precision A series of experiments is conducted to test whether PIS$_1$ can improve retrieval precision (both MAP and P10). This section presents results in tables, first for short queries, and then for long queries. (Table 7.6, page 111 shows the average query length.) The result tables should be read as follows:

- each table is split into an upper and a lower part:
  - the upper part reports on Disks 4&5;
  - the lower part reports on WT10G;
- the rows are arranged in this order:
  - row number (to facilitate referring to the tables);
  - default settings for BM25 and then PL2;
  - optimised settings for BM25 and then PL2;
- the columns are arranged in this order:
  - columns 1 and 2 show the settings and then the measure;
  - column 3 shows the retrieval score of the baseline (=only the retrieval model);
  - column 4 shows the retrieval scores of the first integration of PIS_1 into the retrieval model (Section 7.3.3.4.3);
  - column 5 shows the retrieval score of the second integration of PIS_1 into the retrieval model (Section 7.3.3.4.3).

Tables 7.11 and 7.12 show the results for short and long queries, respectively. The best score(s) for each row is printed in bold. The asterisk * (**) shows statistical significance at p <0.05 (p <0.01), according to the Wilcoxon matched-pairs signed-ranks test.

Tables 7.11 and 7.12 show that:

1. **1st versus 2nd integration of PIS_1 into the model:** retrieval performance with PIS_1 into the retrieval model is better for the second integration (altering the output of the model) than the first integration (altering the input of the model);
2. **short versus long queries:** retrieval performance with PIS_1 into the model is better for long queries than short queries;

Short Queries - Retrieval Precision					
Disks 4&5					
row	settings	measure	baseline	$f_{t,d} \times PIS_1$	$w_{t,d} \times PIS_1$
1	def. BM25	MAP	**0.242**	0.228 (-5.8%)	**0.242 (none)**
2		P10	**0.424**	0.417 (-1.7%)	**0.424 (none)**
3	def. PL2	MAP	0.256	0.253 (-1.2%)	**0.259 (+1.2%)**
4		P10	0.445	0.441 (-0.9%)	**0.447 (+0.4%)**
5	opt. BM25	MAP	**0.254**	0.237 (-7.2%)	**0.254 (none)**
6		P10	0.438	0.428 (-2.3%)	**0.442 (+0.9%)**
7	opt. PL2	MAP	0.257	0.254 (-1.2%)	**0.259 (+0.8%)**
8		P10	0.446	0.441 (-1.1%)	**0.447 (+0.2%)**
WT10G					
row	settings	measure	baseline	$f_{t,d} \times PIS_1$	$w_{t,d} \times PIS_1$
9	def. BM25	MAP	0.187	0.180 (-3.9%)	**0.188 (+0.5%)**
10		P10	0.300	0.304 (+1.3%)	**0.310 (+3.3%)**
11	def. PL2	MAP	0.208	0.206 (-1.0%)*	**0.215 (+3.4%)***
12		P10	0.324	0.320 (-1.3%)*	**0.326 (+0.6%)***
13	opt. BM25	MAP	0.211	0.203 (-3.8%)	**0.212 (+0.4%)**
14		P10	0.328	0.329 (+0.3%)	**0.337 (+2.7%)**
15	opt. PL2	MAP	0.211	0.209 (-1.0%)*	**0.218 (+3.3%)***
16		P10	0.325	**0.326 (+0.3%)***	**0.326 (+0.3%)***

Table 7.11: Retrieval performance with BM25 and PL2 for short queries (WT10G and Disks 4&5).

Long Queries - Retrieval Precision					
Disks 4&5					
row	settings	measure	baseline	$f_{t,d} \times PIS_1$	$w_{t,d} \times PIS_1$
1	def. BM25	MAP	0.242	0.236 (-2.5%)*	**0.256 (+5.8%)****
2		P10	0.423	0.428 (+1.2%)*	**0.436 (+3.1%)****
3	def. PL2	MAP	0.218	0.241 (+10.5%)*	**0.258 (+18.3%)****
4		P10	0.388	0.430 (+10.8%) *	**0.430 (+10.8%)****
5	opt. BM25	MAP	0.244	0.238 (-2.5%)**	**0.259 (+6.1%)****
6		P10	0.423	0.429 (+1.4%)**	**0.437 (+3.3%)****
7	opt. PL2	MAP	0.235	0.241 (+2.6%)*	**0.260 (+10.6%)****
8		P10	0.418	0.430 (+2.9%)*	**0.437 (+4.5%)****
WT10G					
row	settings	measure	baseline	$f_{t,d} \times PIS_1$	$w_{t,d} \times PIS_1$
9	def. BM25	MAP	0.175	0.185 (+5.7%)	**0.200 (+14.3%)****
10		P10	0.334	0.346 (+3.6%)	**0.356 (+6.6%)****
11	def. PL2	MAP	0.167	0.201 (+20.4%) **	**0.215 (+28.7%)****
12		P10	0.320	0.359 (+12.2%) **	**0.360 (+12.5%)****
13	opt. BM25	MAP	0.187	0.200 (+7.0%)	**0.211 (+12.8%)****
14		P10	0.344	0.347 (+0.9%)	**0.362 (+5.2%)****
15	opt. PL2	MAP	0.186	0.201 (+8.1%)	**0.215 (+15.6%)****
16		P10	0.340	0.359 (+5.6%)	**0.360 (+5.9%)****

Table 7.12: Retrieval performance with BM25 and PL2 for long queries (WT10G and Disks 4&5).

3. **MAP versus P10:** P10 seems to be affected from PIS$_1$ slightly less than MAP is affected;
4. **Disks 4&5 versus WT10G:** retrieval performance with PIS$_1$ into the model is better for WT10G than it is for Disks 4&5;
5. **best overall performance:** the best overall performance is always associated with PIS$_1$.

These five observations are generalised over both collections, retrieval models, and retrieval measures. By generalised, we mean that they are mostly consistent, with very few exceptions. Each observation is discussed next.

1. 1st versus 2nd integration of PIS$_1$ into the model The first observation from Tables 7.11 & 7.12 is that integrating PIS$_1$ into the output of the matching process is better for retrieval than integrating it into the input of the matching process. Specifically, the difference in retrieval precision between the two integrations is noted. Regarding short queries:

- **1st integration:** retrieval precision changes from -7.2% (Table 7.11, row 5, column 5) to +1.3% (Table 7.11, row 10, column 5);
- **2nd integration:** retrieval precision varies from no change at all (Table 7.11, rows 1,2,5, last column) up to +3.4% with a statistical significance (Table 7.11, row 11, last column).

Regarding long queries:

- **1st integration:** retrieval precision changes from -2.5% with a statistical significance (Table 7.12, rows 1&5, column 5) to +20.4% with a strong statistical significance (Table 7.12, row 11, column 5);
- **2nd integration:** retrieval precision varies from +3.1% with a strong statistical significance (Table 7.12, row 2, last column) to +28.7% with a strong statistical significance (Table 7.12, row 11, last column).

This difference in retrieval performance between the first and second integration of PIS$_1$ into the retrieval model is not surprising: as discussed in Section 7.3.3.4.3, the matching process is tailored to raw term frequencies, because it normalises them according to document length. PIS$_1$ is not a raw frequency count, but a probability, which does not have to be normalised according to document length. Hence, integrating PIS$_1$ into the output of the matching process is preferred over integrating it into the input of this process.

2. short versus long queries The second observation from Tables 7.11 & 7.12 is that long queries benefit more from PIS_1 than short queries. Specifically, the difference in retrieval precision between the two query types is noted:

- **short queries:** retrieval precision changes from -7.2% (Table 7.11, row 5, column 5) to +3.4% with a statistical significance (Table 7.11, row 11, last column);

- **long queries:** retrieval precision changes from -2.5% with a statistical significance (Table 7.12, rows 1&5, column 5) to +28.7% with a strong statistical significance (Table 7.12, row 11, last column).

This difference in retrieval performance between short and long queries is not surprising: longer queries contain more words, which are not necessarily keywords. Short queries tend to contain few words, which are mainly keywords. Keywords are likely to be informative, hence the contribution of PIS_1 is small. Compared to short queries, long queries tend to contain more terms, which are not necessarily informative, hence the contribution of PIS_1 is bigger.

3. MAP versus P10 The third observation from Tables 7.11 & 7.12 is that P10 seems to be affected from PIS_1 slightly less than MAP is affected. Specifically, when retrieval with PIS_1 into the model hurts performance, the damage is more for MAP than it is for P10, and also when retrieval with PIS_1 into the model benefits performance, the gain is more for MAP than it is for P10.

When integrating PIS_1 into the model hurts retrieval performance, the difference between MAP and P10 is as follows:

- **MAP:** MAP is hurt from -7.2% (Table 7.11, row 5, column 5) to -1.0% (Table 7.11, rows 11&15, column 5);

- **P10:** P10 is hurt from -2.3% (Table 7.11, row 6, column 5) to -0.9% (Table 7.11, row 4, column 5).

When integrating PIS_1 into the model benefits retrieval performance, the difference between MAP and P10 is as follows:

- **MAP:** MAP improves from +0.5% (Table 7.11, row 9, last column) to +28.7% with a strong statistical significance (Table 7.12, row 11, last column);

- **P10:** P10 improves from +0.2% (Table 7.11, row 8, last column) to +12.5% with a strong statistical significance (Table 7.12, row 12, last column).

Overall, P10 seems to be affected from PIS_1 slightly less than MAP is affected. One reason for this could be that using PIS_1 alters the relevance ranking of documents with respect to a query less at the top ranks (measured by P10), and more at the lower ranks (measured by MAP). Hence, this could indicate that PIS_1 benefits recall slightly more than it benefits precision. (Precision and recall were introduced in Section 2.7.) However, the difference between MAP and P10 is not big enough to conclude that using PIS_1 for retrieval benefits recall more than it does precision.

4. Disks 4&5 versus WT10G The fourth observation from Tables 7.11 & 7.12 is that retrieval performance with PIS_1 into the model is better for WT10G than it is for Disks 4&5. Specifically, regarding short queries, the difference between Disks 4&5 and WT10G is as follows:

- **Disks 4&5:** retrieval precision changes from -7.2% (Table 7.11, row 5, column 5) to +1.2% (Table 7.11, row 3, last column);

- **WT10G:** retrieval precision changes from -3.9% (Table 7.11, row 9, column 5) to +3.4% with a statistical significance (Table 7.11, row 11, last column).

Regarding long queries, the difference between Disks 4&5 and WT10G is as follows:

- **Disks 4&5:** retrieval precision changes from -2.5% with a statistical significance (Table 7.12, rows 1&5, column 5) to +18.3% with a very strong statistical significance (Table 7.12, row 3, last column);

- **WT10G:** retrieval precision changes from +0.9% (Table 7.12, row 14, column 5) to +28.7% with a very strong statistical significance (Table 7.12, row 11, last column).

This difference in retrieval performance between short and long queries could be due to the fact that the baseline model (without PIS_1) gave lower scores for WT10G than it did for Disks 4&5, at all times. Hence, since the baseline performed worse on WT10G, there was more room for improvement on this collection, than there was in Disks 4&5. This improvement was made by using PIS_1. In addition, other parameters that could potentially affect the difference in retrieval performance between Disks 4&5 and WT10G could be the difference in the number of queries used (250 for Disks 4&5 and 100 for WT10G), or the difference in size between Disks 4&5 and WT10G. In particular, the difference in size could affect the computation of PIS$_1$ because PIS$_1$ uses the collection statistics, and larger collections might lead to more accurate computation. However, both of these parameters are problematic: first, assuming that all queries are of approximately equal difficulty, there is no reason why having more queries would lead to worse retrieval performance; second, Section 6.5 showed that the computation of PIS$_1$ is not collection-dependent, hence collection size should not alter PIS$_1$ accuracy.

5. Best overall performance The final observation from Tables 7.11 & 7.12 is that the best overall performance is always associated with PIS$_1$. Specifically, regarding average precision (MAP):

- **short queries:**
  - Disks 4&5: the best overall MAP is 0.259, when PIS$_1$ is integrated into the output of PL2, with both default and optimised settings (Table 7.11, rows 3&7, last column);
  - WT10G: the best overall MAP is 0.218, when PIS$_1$ is integrated into the output of PL2, with optimised settings (Table 7.11, row 15, last column). This score is also statistically significant;
- **long queries:**
  - Disks 4&5: the best overall MAP is 0.260, when PIS$_1$ is integrated into the output of PL2, with optimised settings (Table 7.12, row 7, last column). This score is also statistically very significant;
  - WT10G: the best overall MAP is 0.215, when PIS$_1$ is integrated into the output of PL2, with optimised settings (Table 7.12, rows 11&15, last column). These scores are also very statistically significant.

Regarding early precision (P10):

- **short queries:**
  - Disks 4&5: the best overall P10 is 0.447, when PIS$_1$ is integrated into the output of PL2, with both default and optimised settings (Table 7.11, rows 4&8, last column);

  - WT10G: the best overall P10 is 0.337, when PIS_1 is integrated into the output of BM25, with optimised settings (Table 7.11, row 14, last column);

- **long queries:**
  - Disks 4&5: the best overall P10 is 0.437, when PIS_1 is integrated into the output of BM25 & PL2, with optimised settings (Table 7.12, rows 6&8, last column). These scores are also statistically very significant;
  - WT10G: the best overall P10 is 0.362, when PIS_1 is integrated into the output of BM25, with optimised settings (Table 7.12, row 14, last column). This score is also statistically very significant.

The fact that using PIS_1 in retrieval gives the best retrieval performance at all times confirms the validity of the hypothesis that the proposed non-topical information score, computed from POS n-grams, can be succesfully combined with the topical information score, computed from frequency statistics, to enhance retrieval performance.

From the above five observations, it is concluded that PIS_1 can help retrieval performance. We suggest that integrating PIS_1 into the retrieval model at the output of the matching process is one way of doing this.

The next section tests how robust is the use of PIS_1 to retrieval performance.

7.3.3.5.2 Retrieval robustness The previous section presented the effect of integrating PIS_1 into retrieval for default and optimised settings. This effect was measured in terms of retrieval precision (MAP and P10). This section looks at the effect of integrating PIS_1 into retrieval upon retrieval robustness. The aim is to see if PIS_1 affects retrieval performance in an accidental or consistent (i.e. robust) way. For example, we want to test if the +28.7% improvement[1] in MAP reported in Table 7.12, page 127, is a one-off, or a reliable indication of retrieval improvement. This is done by looking at retrieval precision across the range of all parameter values used, not only the default and optimal values presented in the previous section. The resulting MAP and P10 scores are plotted in Figures 7.1 - 7.8, pages 134 - 141. Also, we look at the flatness of MAP and P10 distribution across the range of parameter values, using the spread measure (Equation 7.9, page 124). Smaller spread S indicates less sensitivity to parameter tuning, hence

[1] Improvement in MAP for PL2 on default settings for long queries, Table 7.12, page 127.

Short Queries - Parameter Spread				
Disks 4&5				
settings	measure	baseline	$f_{t,d} \times PIS_1$	$w_{t,d} \times PIS_1$
BM25	MAP	0.264	**0.249**	0.260
	P10	0.816	0.796	**0.786**
PL2	MAP	**0.278**	0.457	0.286
	P10	0.388	0.500	**0.326**
WT10G				
settings	measure	baseline	$f_{t,d} \times PIS_1$	$w_{t,d} \times PIS_1$
BM25	MAP	**0.543**	0.605	0.557
	P10	0.916	**0.896**	**0.896**
PL2	MAP	**0.353**	0.455	0.403
	P10	0.388	0.500	**0.326**

Table 7.13: Robustness of retrieval performance for short queries.

Long Queries - Parameter Spread				
Disks 4&5				
settings	measure	baseline	$f_{t,d} \times PIS_1$	$w_{t,d} \times PIS_1$
def. BM25	MAP	0.694	0.608	**0.490**
	P10	0.811	0.783	**0.618**
def. PL2	MAP	0.581	**0.352**	0.490
	P10	0.814	**0.409**	0.526
WT10G				
settings	measure	baseline	$f_{t,d} \times PIS_1$	$w_{t,d} \times PIS_1$
def. BM25	MAP	0.541	0.484	**0.342**
	P10	0.820	**0.600**	0.660
def. PL2	MAP	0.497	0.296	**0.273**
	P10	0.740	0.420	**0.370**

Table 7.14: Robustness of retrieval performance for long queries.

more robust models. The spread results are presented in Tables 7.13 and 7.14. For each row, smaller spread is printed in bold.

Figures 7.1 - 7.8 plot the parameter value (x axis) and the corresponding MAP or P10 score (y axis), for all experiments in this section. These figures show that the plots of the baseline and the two integrations are similar in shape. Also, the second integration of PIS$_1$ (to the output of document-query matching) is always better than the first integration of PIS$_1$ (to the input of document-query matching), and also better than the baseline (no PIS$_1$ at all). This is consistent for all collections. Hence, integrating PIS$_1$ into the retrieval process affects retrieval performance in a generally consistent way.

Tables 7.13 and 7.14 show the difference between the highest and lowest MAP

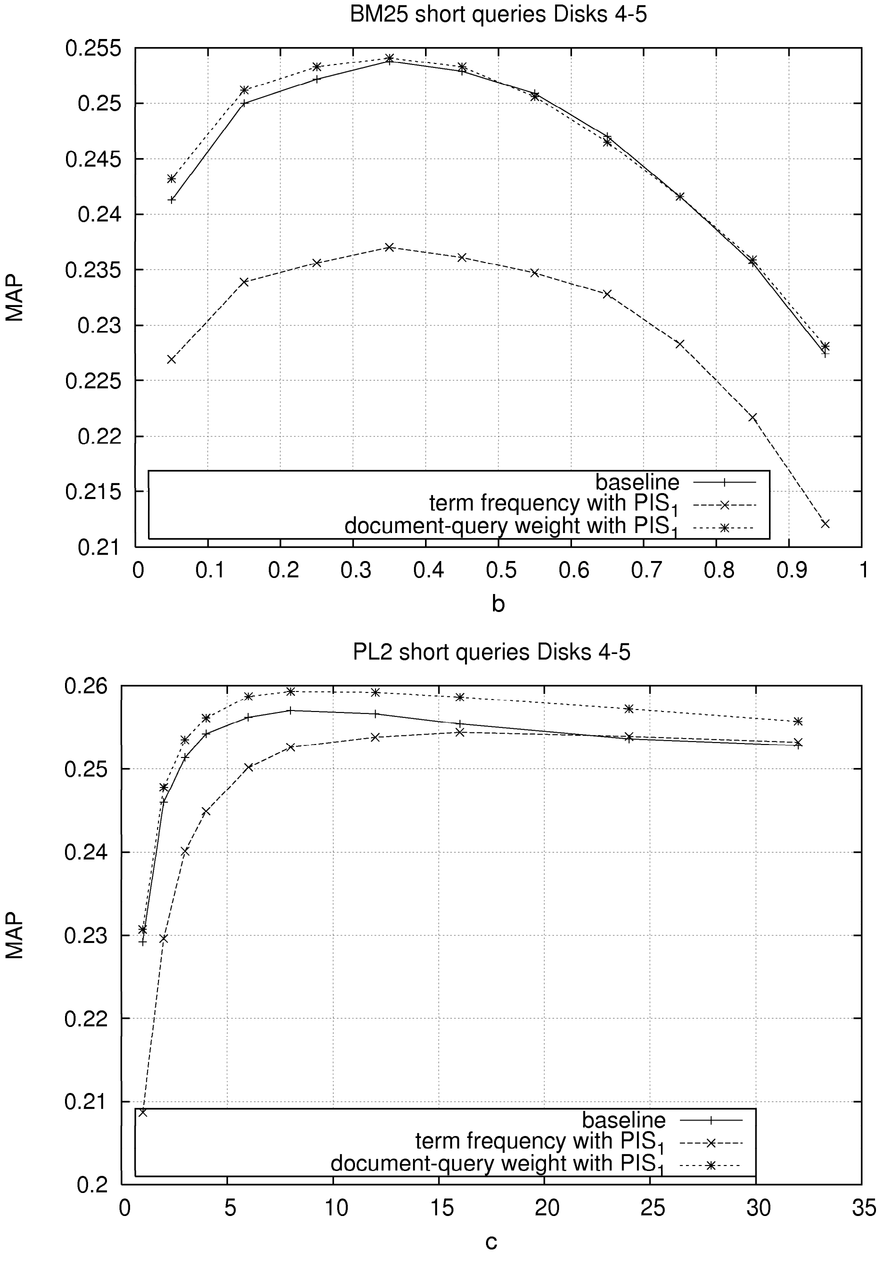


Figure 7.1: Tuning parameters for MAP with and without PIS_1 for short queries.

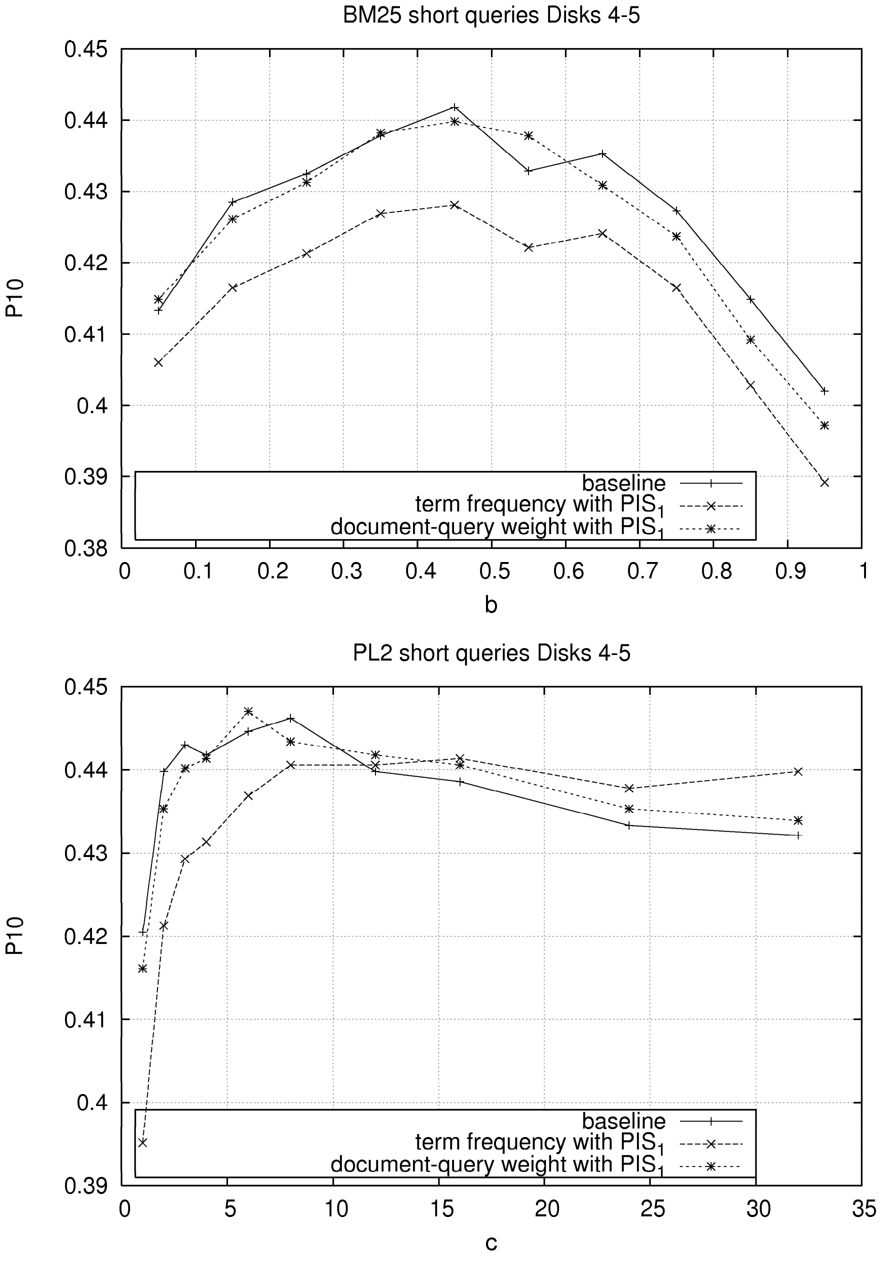


Figure 7.2: Tuning parameters for P10 with and without PIS_1 for short queries.

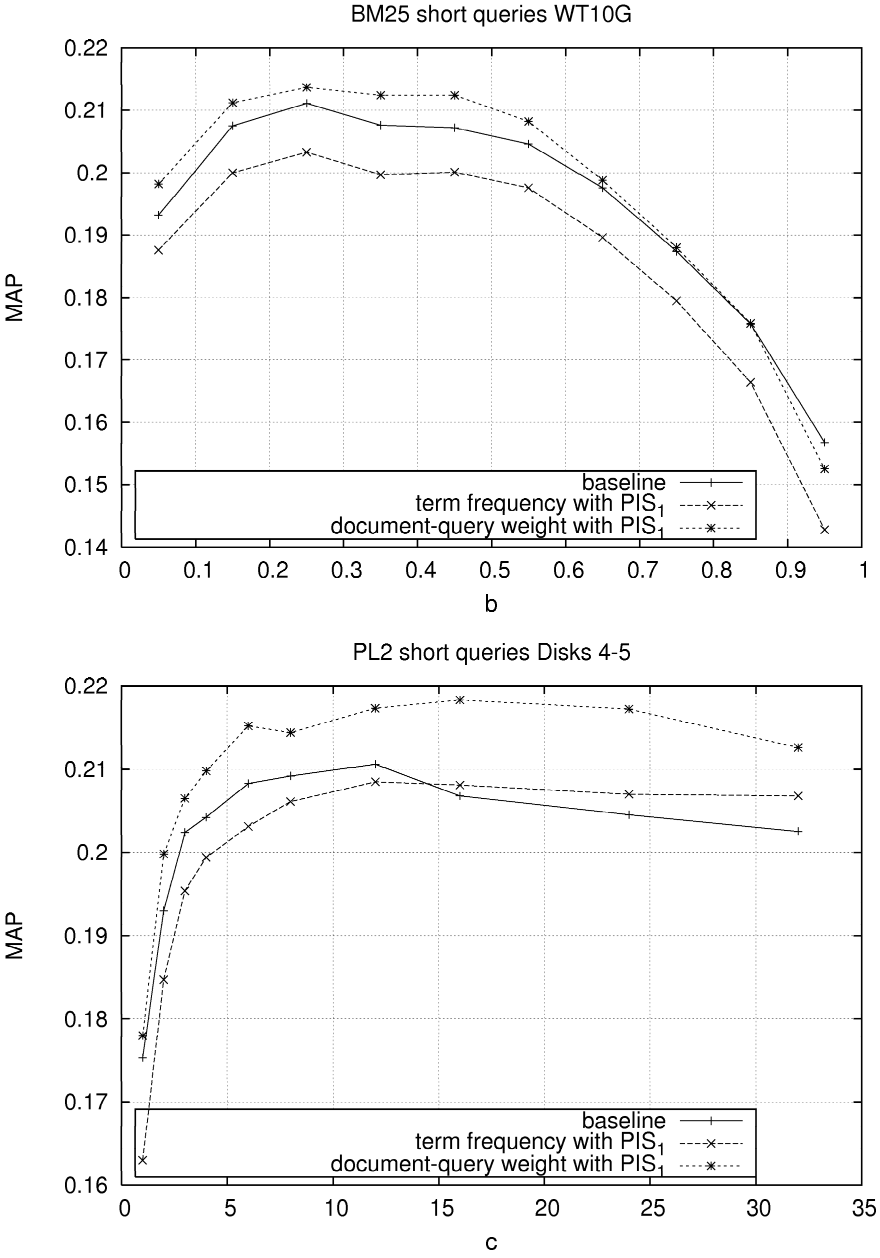


Figure 7.3: Tuning parameters for MAP with and without PIS_1 for short queries.

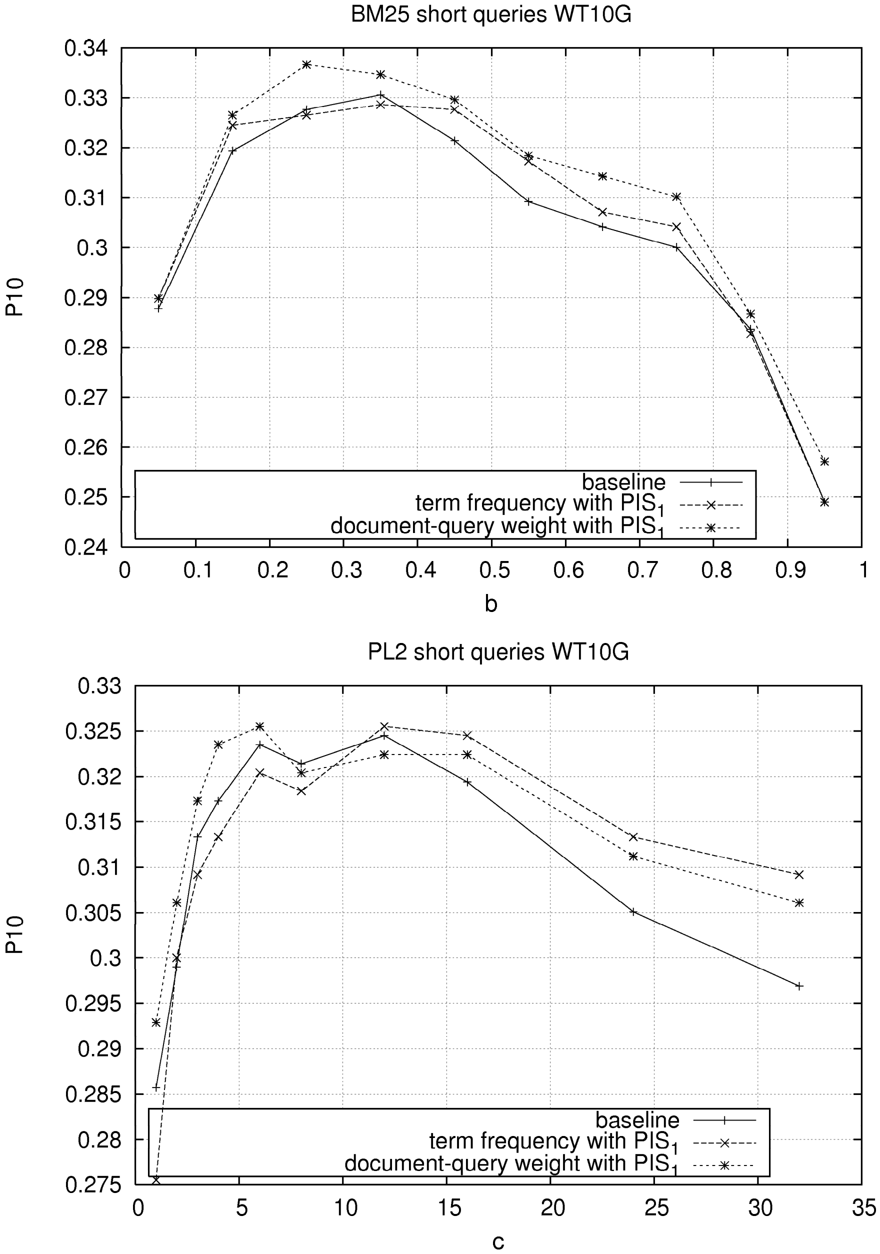
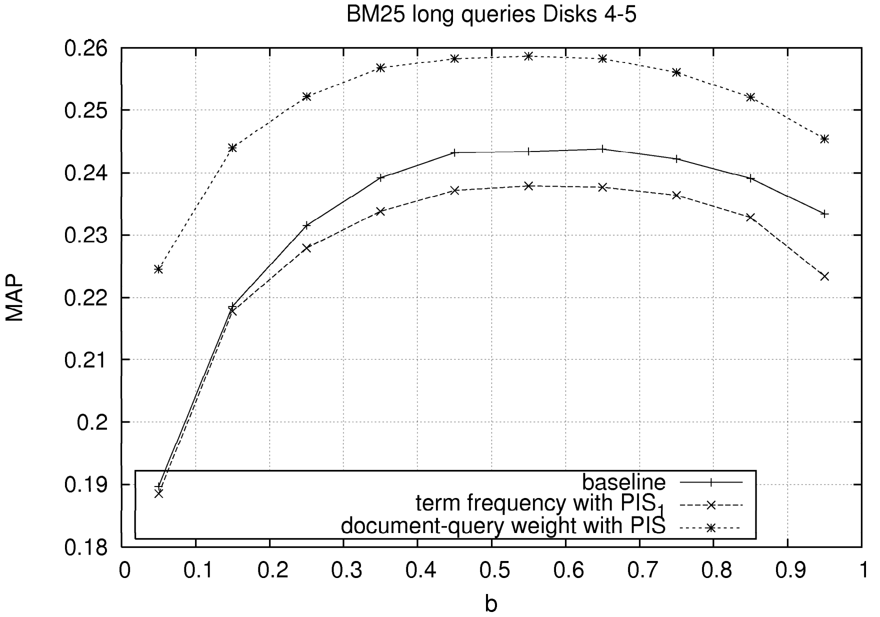


Figure 7.4: Tuning parameters for P10 with and without PIS_1 for short queries.



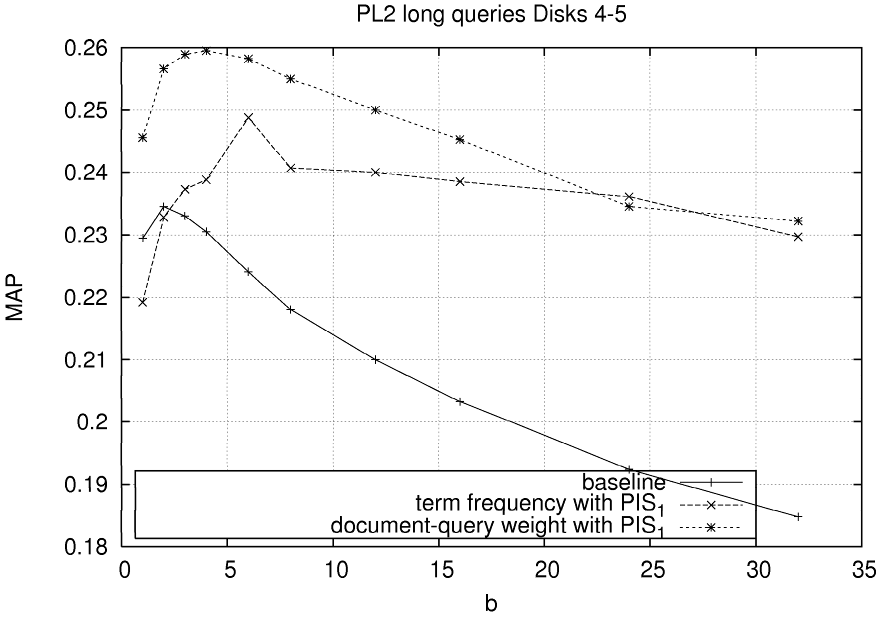


Figure 7.5: Tuning parameters for MAP with and without PIS_1 for long queries.

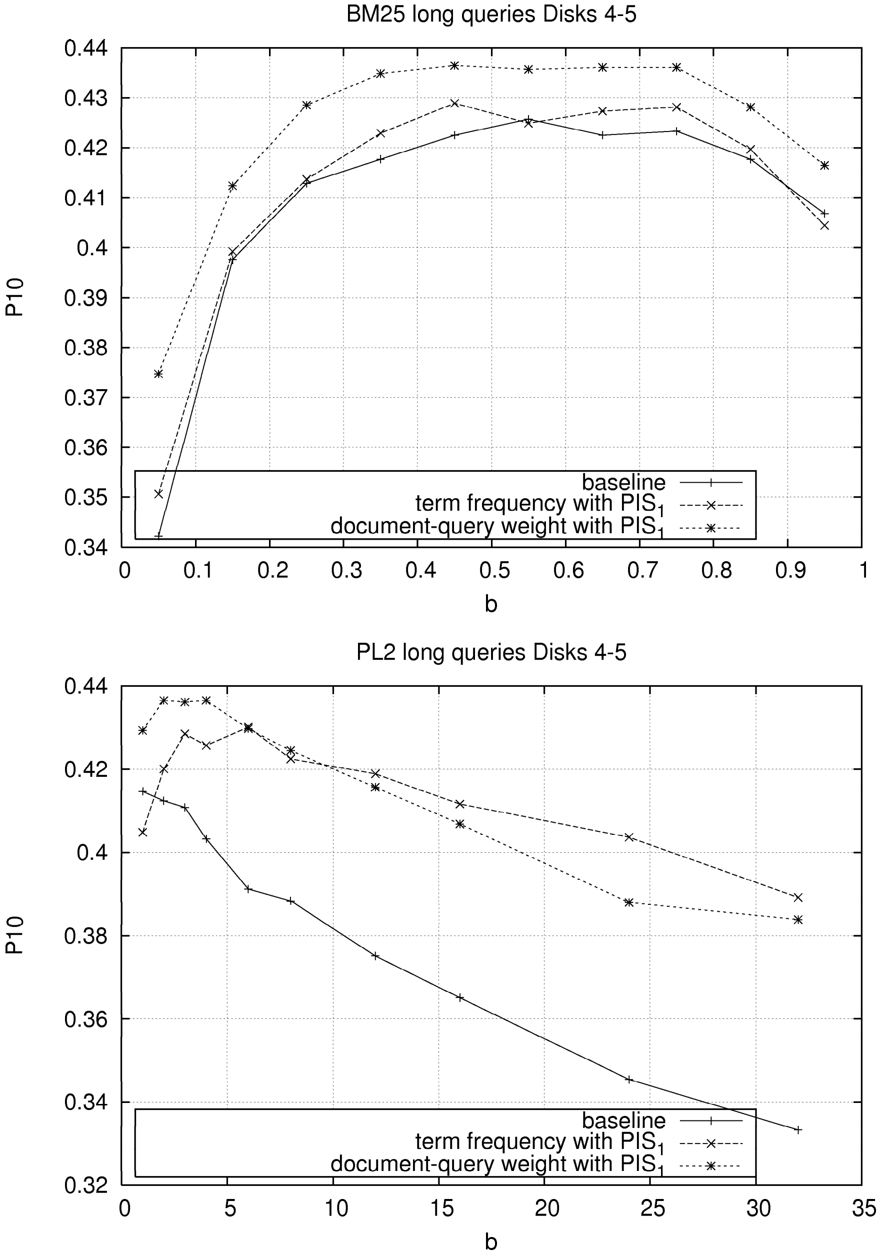


Figure 7.6: Tuning parameters for P10 with and without PIS$_1$ for long queries.

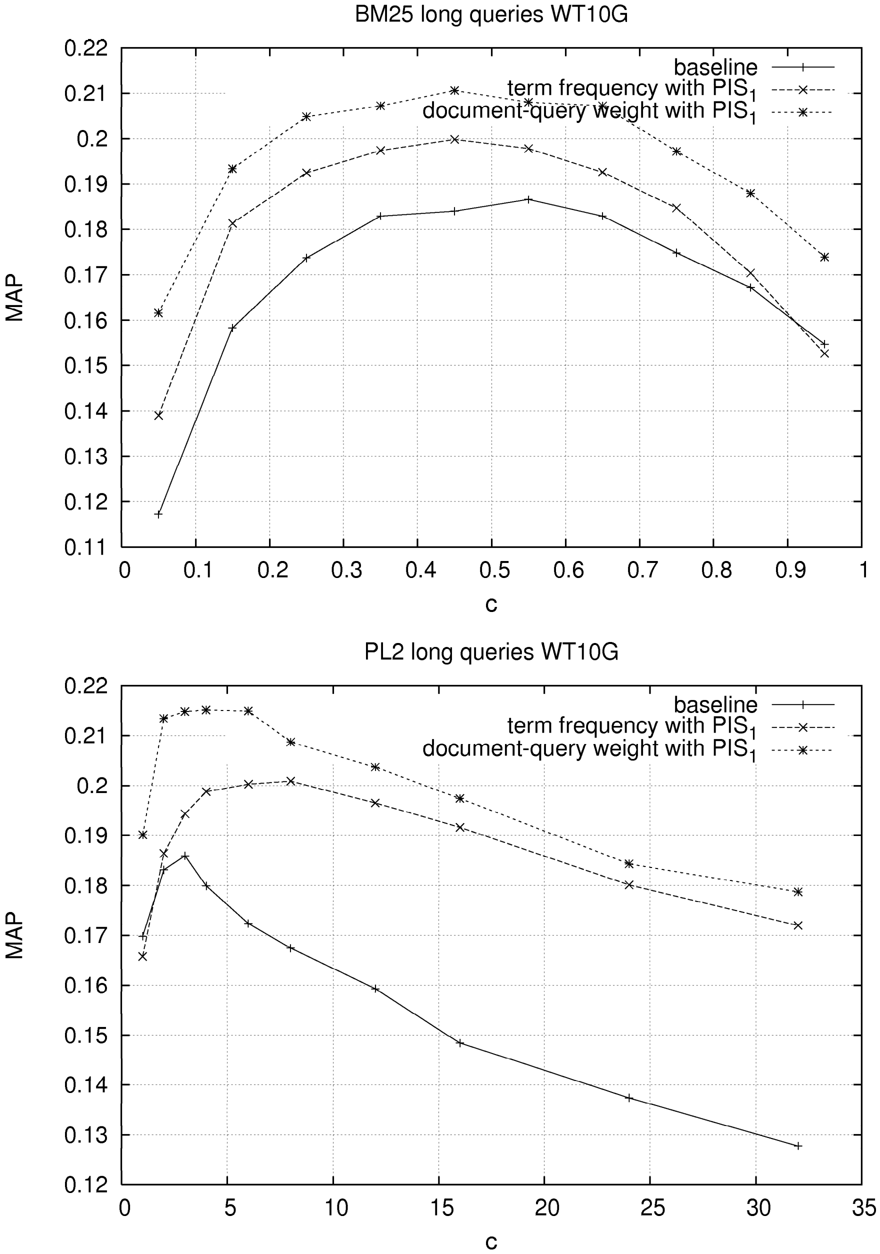


Figure 7.7: Tuning parameters for MAP with and without PIS_1 for long queries.

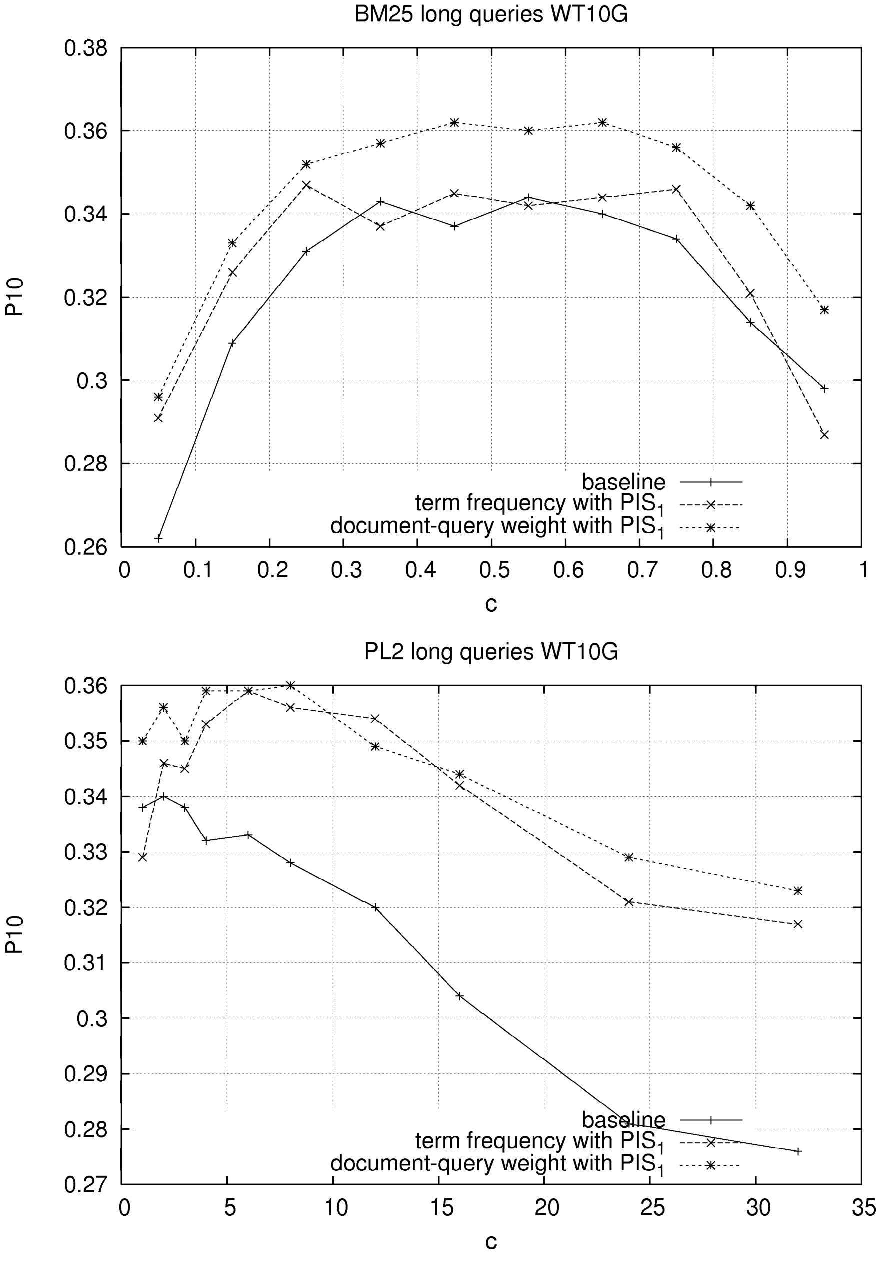


Figure 7.8: Tuning parameters for P10 with and without PIS_1 for long queries.

and P10 across the range of all parameters tried, for short and long queries respectively. This difference corresponds to the highest and lowest peak of each plot in Figures 7.1 - 7.8. The smaller this difference, the more robust the retrieval performance. Overall, the spread values tend to be similar. For short queries, the baseline has the most stable retrieval performance. For long queries, the integration of PIS_1 into the output of the matching process (second integration) has the most stable retrieval performance. These observations are consistent for both collections and evaluation measures.

Overall, integrating PIS_1 into the retrieval model does not render the model less robust and performs in a consistent way. The second integration, which was shown to be most beneficial to retrieval precision in Section 7.3.3.5.1, also gives the most robust performance (lowest spread scores). On the basis of this, it is concluded that PIS_1 can benefit retrieval performance in a robust way.

7.3.3.6 Conclusion

Section 7.3.3 tested the hypothesis that PIS_1, which is derived from POS n-grams, can be combined with the topical content of terms, computed by typical term weighting, to improve retrieval performance. Two ways were suggested for integrating PIS_1 into the retrieval model that matched documents to queries, first into the input of the model, and second into the output of the model. Experiments with established and standard models and datasets showed that PIS_1 can improve retrieval performance robustly across a range of default and optimised settings.

7.4 Summary

This chapter showed that the frequency of POS n-grams in a collection and also the term information score (PIS) computed from POS n-grams can be used successfully by IR systems. Four IR applications of POS n-gram frequency and PIS were suggested. Section 7.2 showed that the frequency of POS n-grams in a collection can be used to detect and remove content-poor text from queries and documents, with benefits to IR performance. Section 7.3.2 showed that PIS is comparable to the established inverse document frequency (IDF) term weight. Section 7.3.3 suggested how PIS can enhance an already competitive retrieval performance, in a robust way.

The next chapter summarises the contributions and conclusions of this thesis, and suggests future research directions.

Chapter 8

Conclusions

8.1 Contributions and conclusions

This thesis investigated the use of part of speech (POS) n-grams, as a representation of shallow grammatical and structural information in language, to Information Retrieval (IR). Based on the empirical finding that there exists a relation between POS n-gram frequency and informative content, a framework was introduced for deriving a weight of non-topical informative content for words from POS n-grams, which was called part of speech information score (PIS). Applications of POS n-gram frequency and also of PIS to IR were presented.

This section summarises the conclusions and contributions of this thesis, and suggests future research directions.

8.1.1 Contributions

The main contributions of this thesis are the following.

- It used a linguistic theory for ranking parts of speech, namely Jespersen's Rank Theory, in IR. To our knowledge, this is the first time that this theory is used in IR or any other automatic language processing technology.

- It presented heuristical evidence suggesting that there exists an approximately directly proportional relationship between POS n-gram frequency and informative content. This novel finding is the opposite of what is observed with words, for which the relationship between frequency and informative content is approximately inversely proportional.

- It introduced a framework for deriving an original term information score exclusively from POS n-grams, based on the relationship between POS n-gram frequency and informative content and also on Jespersen's Rank Theory.

- It used POS n-grams not as a feature for classification, neither to make predictions about the occurrence of parts of speech/words, as has been done so far, but as a feature of non-topical informative content. This is a novel use of POS n-grams.

- It examined the statistical properties of POS n-grams and of the proposed term information score that is computed from them in a series of thorough and unbiased experiments, which included five standard and established collections of different size (totalling >32GB) and domain, three established state of the art POS taggers, and a variation of the n-gram order n between n= 1 - 100. Experimental evidence showed that POS n-grams are distributed similarly in different collections, and that the POS n-gram based term information score is positively correlated to inverse document frequency.

- It suggested four novel applications of POS n-grams to IR and evaluated them on standard and established datasets, under default and competitive settings. Experimental evidence showed that retrieval performance enhanced considerably.

8.1.2 Conclusions

This section discusses the achievements and conclusions of this thesis.

8.1.2.1 Part of speech n-grams and informative content

Traditionally, n-grams are extracted from contiguous sequences, hence they are themselves contiguous subsequences. The ordering of their components has been manipulated extensively, mainly to make predictions about the occurrence of a component, or to characterise the sequences from which they were extracted. This thesis uses POS n-grams in a different and novel way. Specifically, it manipulates

- the linguistic properties of their components, namely parts of speech, for which some basic hierarchy stands (Jespersen's Rank Theory). This hierarchy is extended to indicate the presence or absence of non-topical content;

- the frequency of the n-grams in a large sample. Based on empirical evidence that links POS n-gram frequency to informative content, POS n-gram frequency is used to quantify informative content.

In a simple manner, this thesis combines these two facts, and looks at POS n-grams in a different light: as strings of things for which we have some prior knowledge. This is a novel representation, and this thesis shows that it is empirically valid and also beneficial to IR applications.

In addition, this thesis presents empirical evidence which indicates that there exists an approximately directly proportional relationship between the frequency and informative content of POS n-grams, contrarily to words, for which frequency and informative content are approximately inversely proportional. This novel finding, combined with the fact that POS n-grams tend to be generally distributed in a Zipfian manner and also similarly in different collections, have allowed for the development of different and successful applications of POS n-grams to IR.

8.1.2.2 Non-topical term weight

One of the applications of POS n-grams to IR presented in this thesis is the derivation of a non-topical term weight using exclusively POS n-grams, called PIS. This term weight is novel both with respect to its derivation from POS evidence only, and also because it is *non-topical*: more specifically, whereas conventional IR systems retrieve information assumed to be relevant to some point of reference, e.g., a query, and as such the rely on *topical* information, this thesis presents a term weight of non-topical information, and shows that its combination to existing topical term weights is useful to IR.

A general methodology is presented for computing this non-topical term weight from POS n-grams, which makes use of a linguistic theory for ranking parts of speech and is implemented using probabilistic approximations, but is not bounded by them, i.e., different linguistic theories or mathematical approximations could be used to produce further variants of this non-topical information score. This point is discussed as a direction of future research in Section 8.2.

Finally, the non-topical term weight presented in this thesis is examined alongside the established inverse document frequency (IDF) term weight, and a series of similarities and differences are found between the two. There are three main differences between PIS and IDF:

- IDF approximates the power of a term in discriminating between documents, whereas PIS approximates the non-topical informative content in a term, regardless of how many documents the term occurs in.
- IDF uses lexical statistics (word/document counts), whereas PIS uses shallow grammarical statistics (POS n-grams).
- IDF is a bag-of-words measure (because it does not consider term context), whereas PIS considers the 'part of speech context' of a term.

There are two main similarities between PIS and IDF:

- It can be argued that both computations (for IDF and PIS) use similar methodologies but different ingredients (term statistics in IDF, POS n-gram statistics in PIS). On one hand, the intuition behind IDF is that a word occurring in many documents is not likely to be very informative. On the other hand, PIS looks at how many POS n-grams in a collection 'contain' a word - ('contain' = map to word n-grams that contain that word), on the intuition that a word occurring in many *and* informative POS n-grams is likely to be informative.
- The second similarity is that IDF and PIS are found to be correlated, and also approximately equally beneficial to IR systems. This similarity is particularly noted for one of the two alternative computations of PIS proposed in this thesis, namely PIS_2, because PIS_2 uses a POS n-gram statistic which resembles term frequency (which is used in IDF implicitly) very closely: specifically, PIS_2 uses the frequency of POS n-grams which 'contain' a term in a collection, and this frequency may be seen as an approximation of the actual term frequency in the collection.

In light of the above differences and similarities, PIS may be seen as an 'approximation' of IDF, from a completely different linguistic angle.

8.2 Future work

This section suggests how parts of this thesis can be extended in the future.

8.2.1 Primary parts of speech

This thesis investigates the use of POS n-grams to IR, when using only the primary parts of speech of language, namely the 14 classes shown in Table 3.1, page 40. This restriction of part of speech classes into 14 categories only has been motivated by efficiency and scalability concerns, and has been shown to be very successful for the applications to IR systems presented in this thesis. Nevertheless, more part of speech categories can also be used, especially with applications that could benefit from the extra granularity in the linguistic analysis introduced by more fine-grained distinctions between individual parts of speech. For instance, in selective applications of query expansion or question answering, it could be useful to distinguish between proper nouns and common nouns, especially as proper nouns are popular queries submitted by Web users to IR systems. Additionally, domain-specific applications, for instance geographical IR or legal IR, could benefit considerably from a classification that would distinguish between certain types of prepositions (e.g., when giving travelling directions, the semantic difference between the prepositions `from` and `to` is of utmost importance; similarly, in legal IR, the semantic difference between the prepositions `for` and `against` is very important.) Currently, such POS evidence is used in an ad-hoc and mostly heuristical manner by domain-specific IR applications. This thesis has introduced a general framework for using POS n-grams, in which such POS evidence could be incorporated simply by adding another POS class to the initial category of POS n-grams, or even by over-riding existing linguistic classifications of parts of speech and defining 'new classes' on demand (e.g., by creating a special POS 'tag' for prepositions of interest to a domain, and by grouping all other prepositions under a single category). The use of POS n-grams, where such individual POS classes are properly defined is an interesting area of future research which could benefit IR technology.

8.2.2 Informative content in part of speech n-grams

Section 5.2 presented heuristical evidence suggesting that there exists a relationship between the frequency and informative content of POS n-grams. The informative content of POS n-grams was computed using heuristics, specifically Algorithms 3 & 4, both of which produced a 'score' of informative content for POS n-grams by:

- rewarding the presence of content-rich individual parts of speech inside the n-gram, and

- ignoring or penalising the presence of content-poor individual parts of speech inside the n-gram.

Algorithms 3 & 4 are not the only way of estimating the informative content of POS n-grams. Other variations can also be used, for instance to reward or penalise parts of speech on an individual basis, not only according to Jespersen's Rank Theory as used in this thesis, but also according to more fine-grained rankings of parts of speech that may be motivated from other linguistic theories (less crude[1], or which incorporate additional linguistic evidence, such as syntactic rules or semantic ontologies). Alternatively, individual parts of speech can be rewarded or penalised according to statistical evidence about their frequency and occurrence in a collection.

This is an interesting area of future investigation, in which there is not much research at the moment. In fact, apart from the work described in this thesis, there is no other literature -to our knowledge- which investigates the informative content of POS n-grams. Development in this line of work is interesting not only to IR, where it would improve the applications presented in this thesis, but also to other automatic language processing fields, like automatic summarisation or extraction of text for instance, where more accurate approximations of the informative content of POS n-grams could produce non-topical (hence fast) approximations of what are the most salient parts of some text. The width of these parts could be easily tuned by varying the order of POS n-grams, in order to customise the compression of the resulting summaries and extracts.

8.2.3 Part of speech probabilities from statistics

Chapter 5 computed the informative content for POS n-grams, from which a non-topical content for words was then derived. This derivation was based on certain assumptions, which, even though motivated from Jespersen's Rank Theory, may be seen as crude approximations, and hence could de further refined. One such explicit assumption was that closed class parts of speech are always non-informative, and that open class parts of speech are always informative. Another assumption

[1] The reason why Jespersen's Rank Theory is considered crude was discussed in Section 3.3.

was that verbs, adjectives and participles are always equally informative, and always less informative than nouns. These assumptions could be replaced by better approximations, which either make more mathematically accurate computations of informative content on the basis of POS statistics, or implement more refined linguistic formalisms, such as the syntactic roles, discourse structure or semantic ontologies corresponding to individual parts of speech. This line of work could lead to further and more refined computations of non-topical term weights. Given the success of the proposed non topical term weight to IR presented in this thesis, it would be worth applying such more refined term weights to IR.

8.2.4 Alternative combinations of probabilities

Chapter 5 presented a methodology for computing the non-topical informative content of terms from POS n-grams. The computation of this term score was realised as a combination of probabilities, and, for this thesis, this combination was realised linearly, by simple addition. Alternative combinations of probabilities may also be used without affecting the general methodology for computing the term score. Specifically, in Equation 5.13, an alternative way to the linear combination of the probabilities $P(inf|pos)$ inside a POS n-gram would be to compute their product or sum their logarithms. Generally, these alternatives are considered approximately equivalent. The linear combination was chosen for simplicity in this thesis: Summation of probabilities is simpler, because multiplication would require smoothing[1], and summation of logarithms would be computationally costly[2]. Another alternative to computing the probability of a POS n-gram being informative would be to view the n-gram as a whole, instead of considering its individual member POS tags. For instance, a POS n-gram pos_j^{j+n-1} can be viewed as a set of pos_j events, and the probability of the POS n-gram being informative can be computed by the probability of its individual POS events occurring in it. For example, using a multinomial model:

$$pos_j^{j+n-1} = pos_j, pos_{j+1}, ..., pos_{j+n-1} \tag{8.1}$$

and then

[1] Without smoothing, a zero probability nullifies the product, and a probability of 1 does not contribute to the product.

[2] Computing logarithms is considered a computationally expensive process.

$$P(inf|pos_j^{j+n-1}) = \frac{P(pos_j^{j+n-1}|inf)P(inf)}{P(pos_j^{j+n-1})} \tag{8.2}$$

$$= \frac{P(pos_j, pos_{j+1}..., pos_{j+n-1}|inf)P(inf)}{P(pos_j, pos_{j+1}..., pos_{j+n-1})} \tag{8.3}$$

Assuming a multinomial model and binary independence between two POS events (pos_j and pos_i) inside a POS n-gram:

$$P(inf|pos_j^{j+n-1}) = P(inf)\prod_{j=1}^{n-1}\frac{P(pos_i|inf)}{P(pos_i)} \tag{8.4}$$

$$= \prod_{j=1}^{n-1} P(inf|pos_i) \tag{8.5}$$

This alternative computation of considering the POS n-gram as a whole is not used in this thesis, because it relies more on statistical information of part of speech occurrence and less on linguistic divisions and ranks of parts of speech. However, this computation is a mathematically attractive alternative, which would be worth investigating in the future with expected benefits to retrieval performance.

8.2.5 Alternative applications of part of speech n-grams

The frequency of POS n-grams and the non-topical term weight derived from them presented in this thesis are not restricted to IR, but can be applied in numerous ways to various applications. For instance, this thesis showed how the proposed term weight can be integrated into retrieval models as another ingredient. Other integrations of this weight into the retrieval model are possible, using language models to match document to queries, for example. Similarly, the proposed term weight is not IR-specific; there is no reason why it cannot be applied to other processing involving term weighting, e.g., automatic summarisation, as discussed above.

Similarly, other applications which do not involve term weighting could also benefit from the proposed term weight. For instance, in POS tagging, it would be interesting to look at the PIS score of ambiguous terms or terms that are consistently tagged erroneously, and see if there is a relation between terms that are often ambiguous or used in an ambiguous way and their non-topical informative content. However, for such an application, because computing PIS already requires POS tagged text, different POS taggers would have to be used.

Another potential future application of PIS could be classification, where it would be interesting to sum the PIS of all terms occurring in a document ('document PIS'), and use it as a classification feature or threshold. For example, what is the average 'document PIS' of technical documents as opposed to Web documents? 'Document PIS' could also be a feature of language comprehension and difficulty, such as the ones used in readability formulae for language teaching (Mikk, 2001; Savicky & Hlavacova, 2002; Zubov, 2004). Furthermore, in machine translation, it would be worth investigating whether PIS (of words or documents) is consistent in parallel text. For instance, when/why does a high-PIS English term translate into a low-PIS German term? Such questions could cast light onto discourse or semantic aspects which are not always easy to handle in machine translation. Finally, in emotion detection, it would be worth looking into possible links between PIS with sentiment or opinion polarity, a popular research area at the moment.

Appendix A

Parts of speech

Tables A.1 & A.2 relate to Chapter 3, and specifically Section 3.2, Table 3.1, page 40. Table A.1 displays the primary and secondary part of speech (POS) categories of the original Penn TreeBank set (Marcus *et al.*, 1993). Table A.2 displays the correspondence between the POS abbreviations of the original Penn TreeBank set and those used in this thesis.

Abbr.	Part of Speech	Abbr.	Part of Speech	Abbr.	Part of Speech
CC	coordinating conjunction	NNS	noun, plural	VBP	verb, non-3rd person singular present
CD	cardinal number	NP	proper noun, singular	VBZ	verb, 3rd person singular present
DT	determiner	NPS	proper noun, plural	WDT	wh-determiner
EX	existential `there`	PDT	predeterminer	WP	wh-pronoun
FW	foreign word	POS	possessive ending	WP$	possessive wh-pronoun
IN	preposition or subordinating conjunction	PP	personal pronoun	WRB	wh-adverb
JJ	adjective	PP$	possessive pronoun	TO	`to`
JJR	adjective, comparative	RB	adverb	UH	interjection
JJS	adjective, superlative	RBR	adverb, comparative	VB	verb, base form
LS	list item marker	RBS	adverb, superlative	VBD	verb, past tense
MD	modal verb	RP	particle	VBG	verb, gerund or present participle
NN	noun, singular or mass	SYM	symbol	VBN	verb, past participle

Table A.1: Primary & secondary part of speech categories (Penn Treebank set).

TreeBank abbr.	**Thesis abbr.**	**Part of Speech**
JJ, JJR, JJS	JJ	adjective
RB,RBR,RBS	RB	adverb
CD, LS	CD	cardinal number
CC	CC	conjunction
DT, WDT, PDT	DT	determiner
MD, VB, VBD, VBG, VBN, VBP, VBZ, VH, VHD, VHG, VHN, VHP, VHZ	MD	auxiliary/modal verb
NN, NNS, NP, NPS, FW	NN	noun
PP, WP, PP$, WP$, EX, WRB	PP	pronoun
IN, TO	IN	preposition
POS	PO	possessive ending
RP	RP	particle
SYM	SY	symbol
UH	UH	interjection
VV, VVD, VVG, VVN, VVP, VVZ	VB	main verb

Table A.2: Part of speech abbreviations in the Penn TreeBank and this thesis.

Appendix B

Part of speech n-grams

Figures B.1 - B.10 relate to Chapter 6. Specifically, Figures B.1 - B.2 refer to Section 6.3.2.2, Figures 6.1 - 6.2, pages 83 - 84, where the distribution of POS n-grams is compared in five different collections. Figures B.3 - B.10 refer to Section 6.3.3, Figure 6.3, page 86. Figures B.3 - B.10 display the plot of POS n-gram frequency in the collection (x axis) versus POS n-gram frequency rank (y axis) for POS n-grams extracted from the AP collection, and for n values between 1 - 100. The AP collection has been presented in Section 6.3.2, Table 6.3, page 80. The collection has been POS tagged with the TreeTagger, presented in Section 3.4.3, page 45.

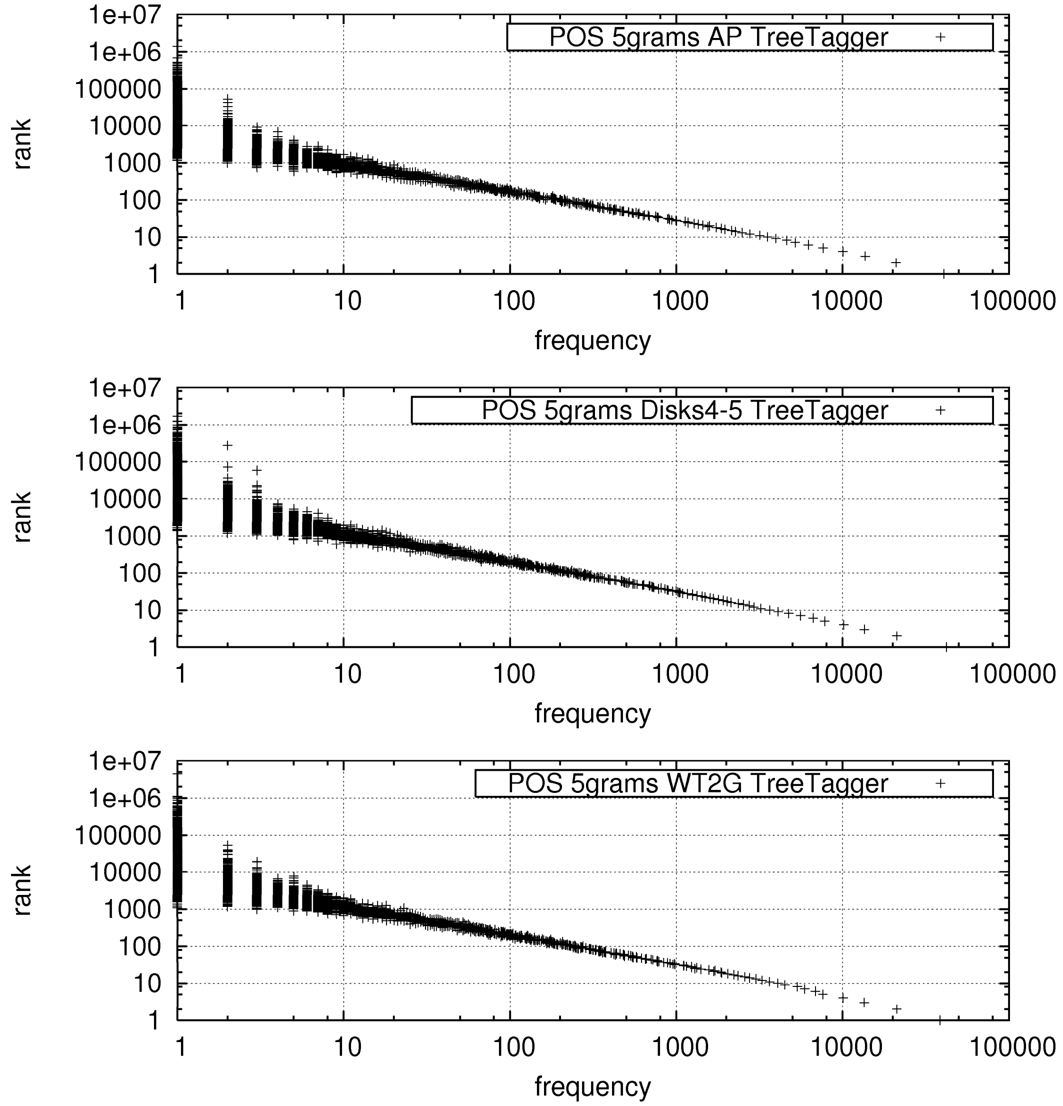


Figure B.1: Distribution of POS 5-grams (AP, Disks 4&5, WT2G).

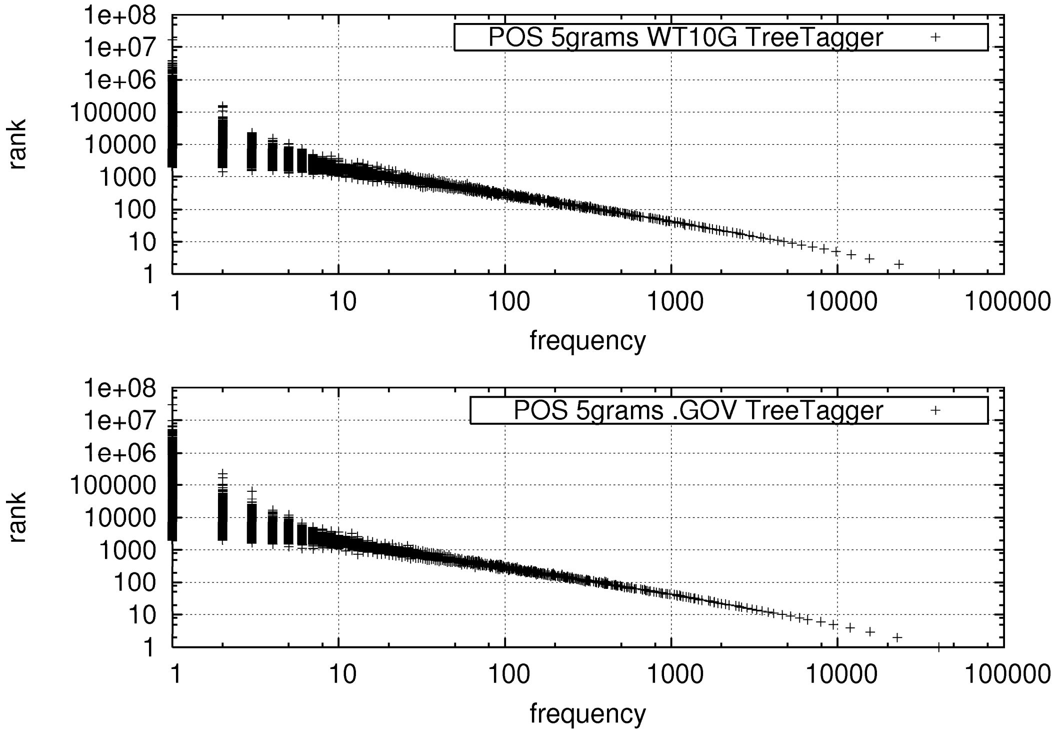


Figure B.2: Distribution of POS 5-grams (WT10G, .GOV).

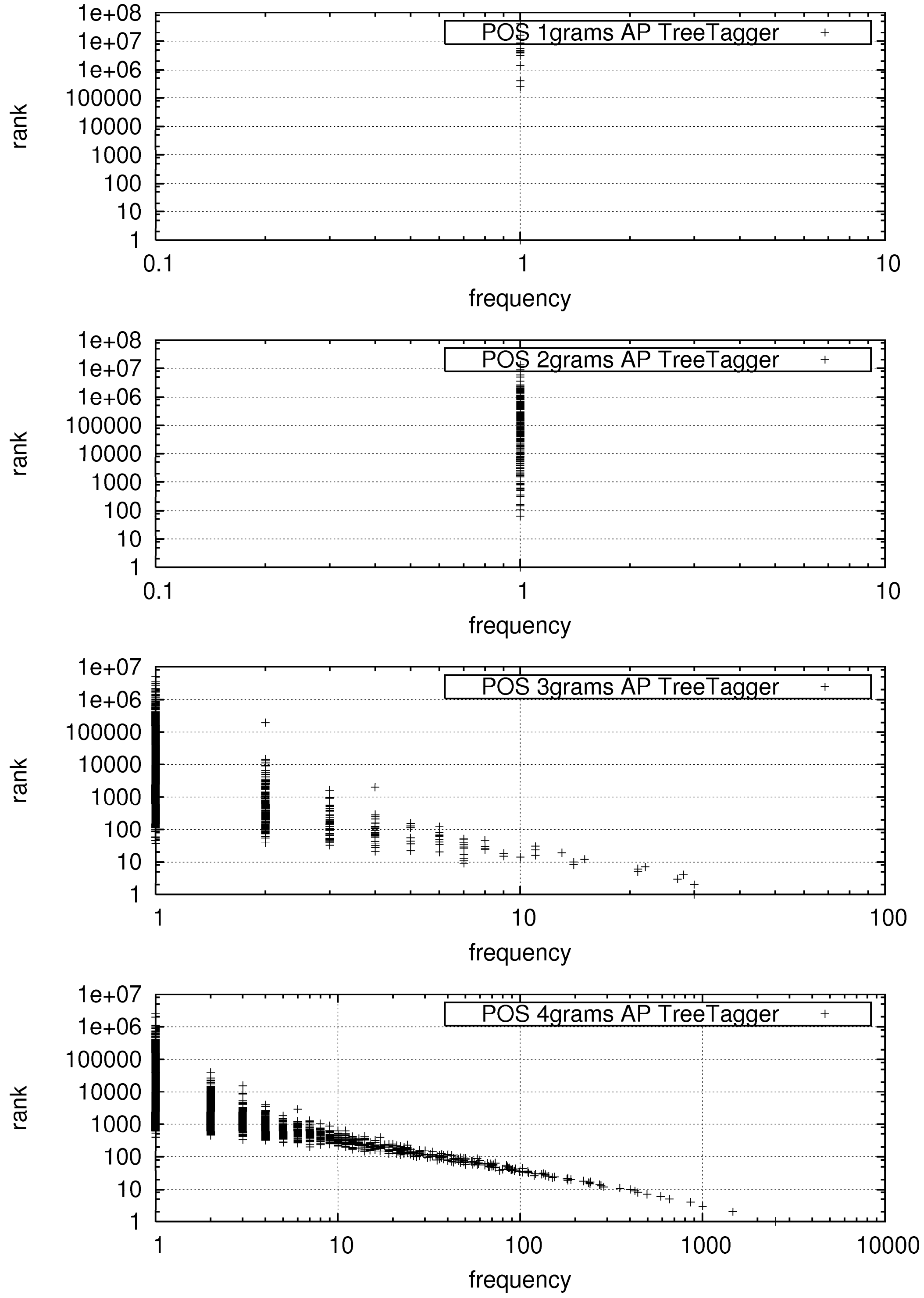


Figure B.3: Distribution of POS n-grams, for n = 1, 2, 3, 4 (AP).

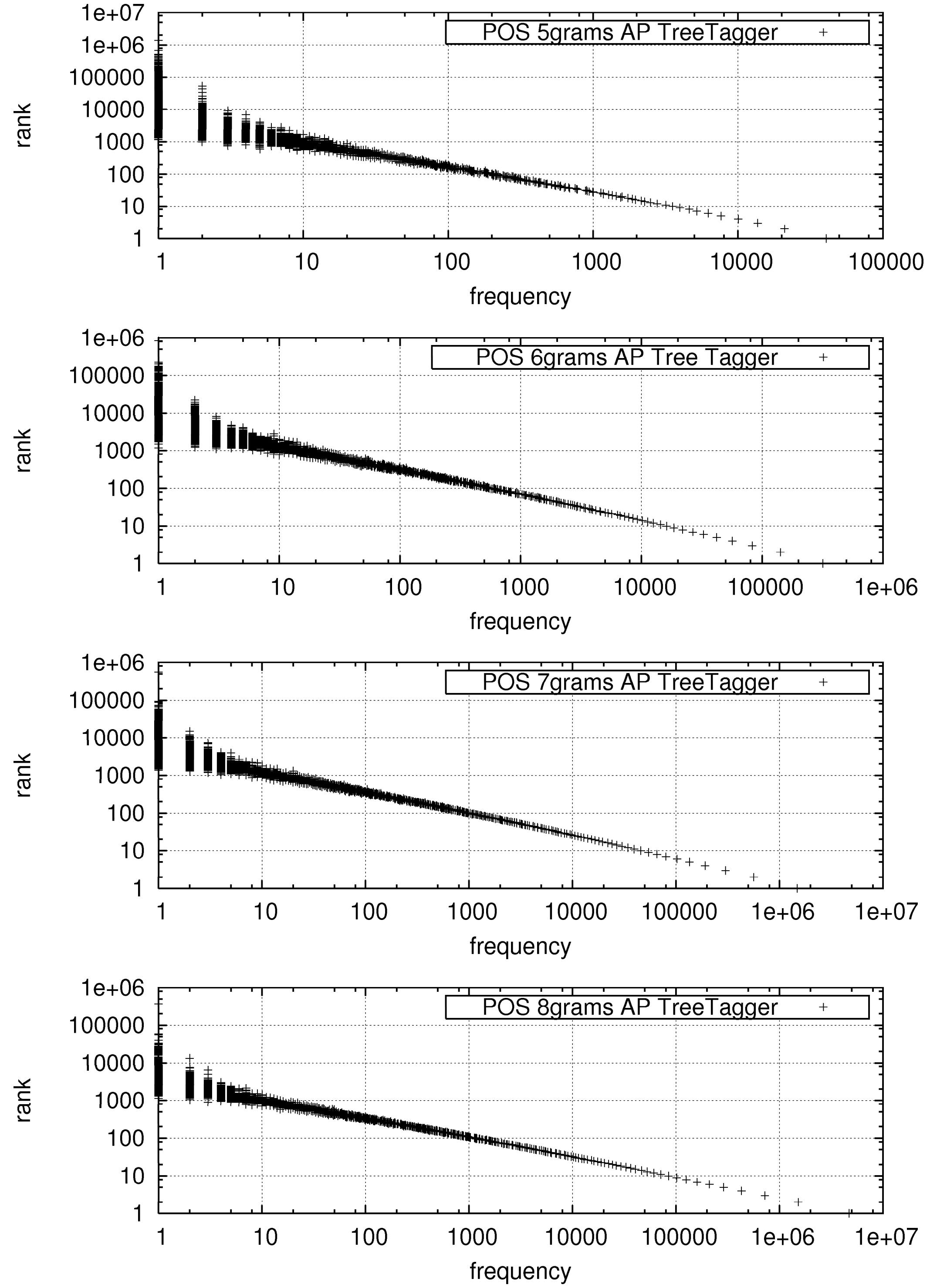


Figure B.4: Distribution of POS n-grams, for n = 5, 6, 7, 8 (AP).

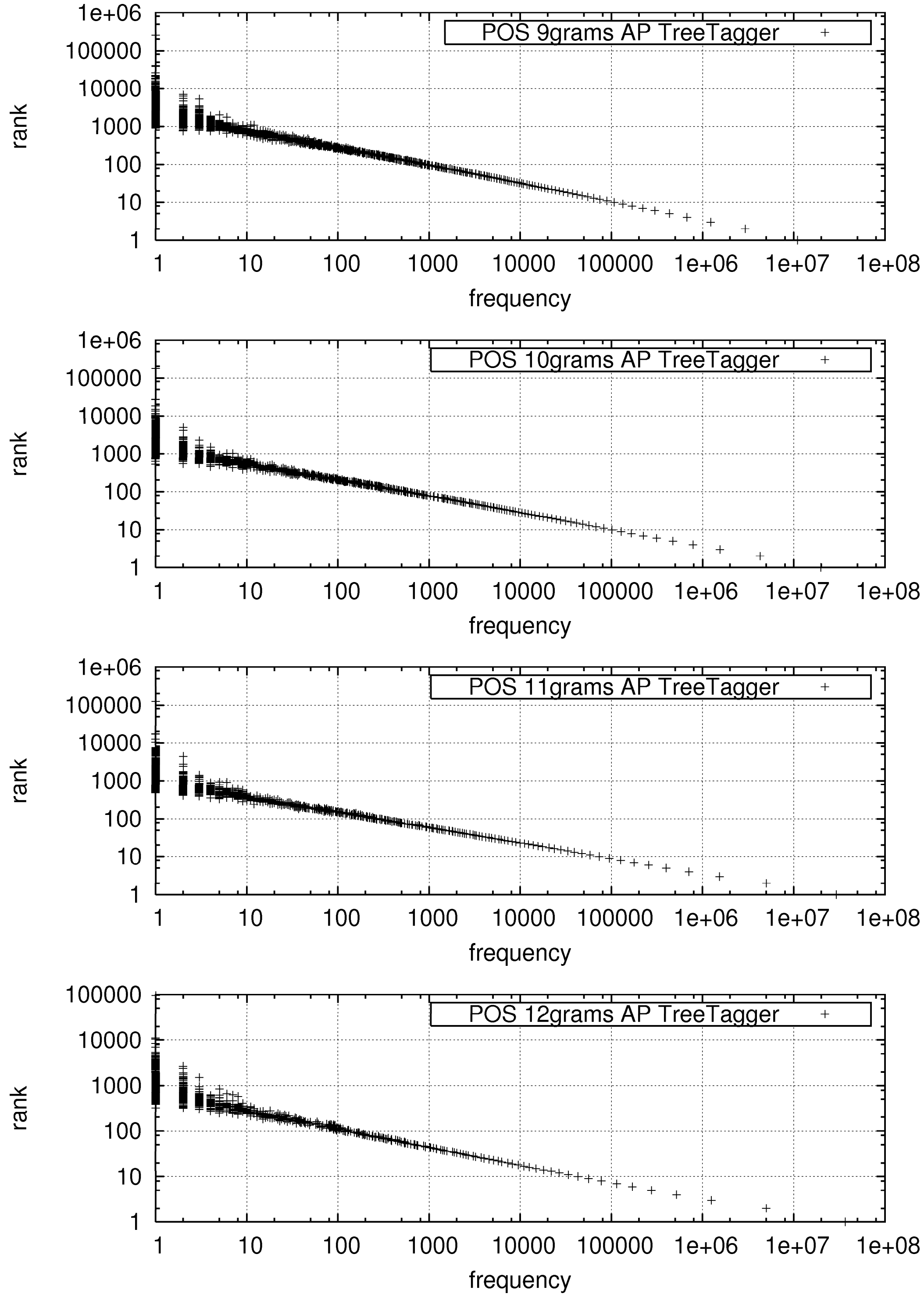


Figure B.5: Distribution of POS n-grams, for n = 9, 10, 11, 12 (AP).

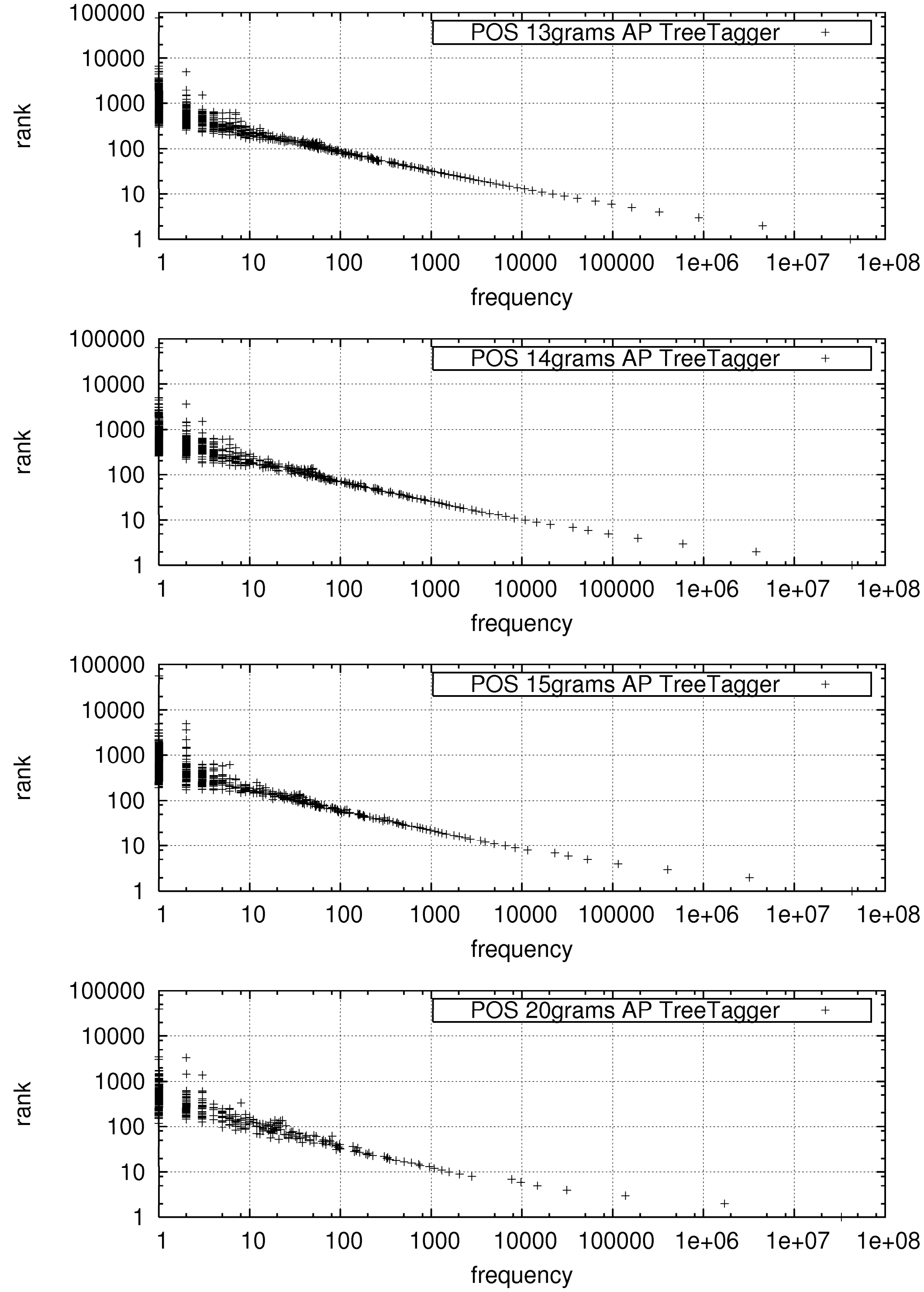


Figure B.6: Distribution of POS n-grams, for n = 13, 14, 15, 20 (AP).

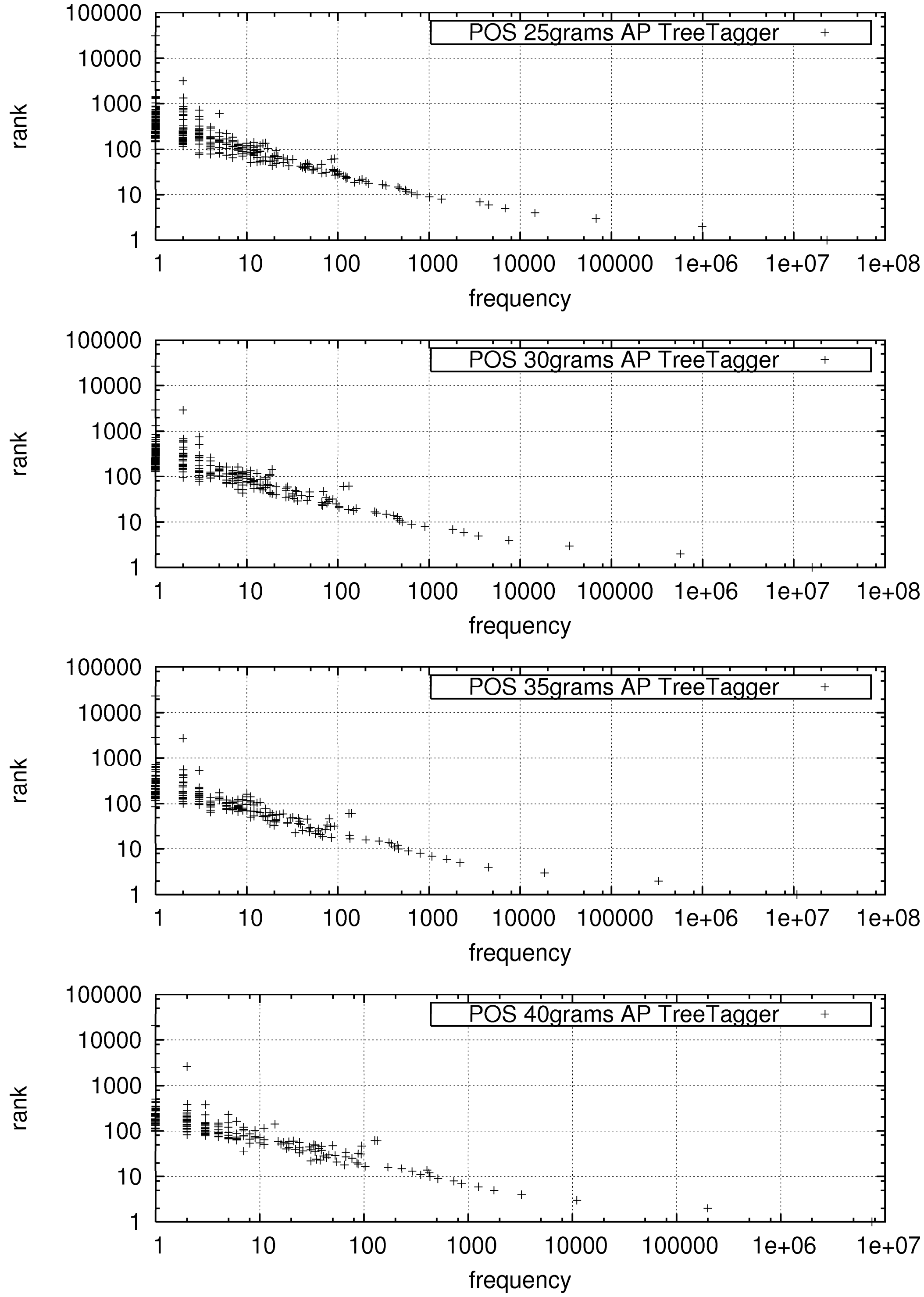


Figure B.7: Distribution of POS n-grams, for n = 25, 30, 35, 40 (AP).

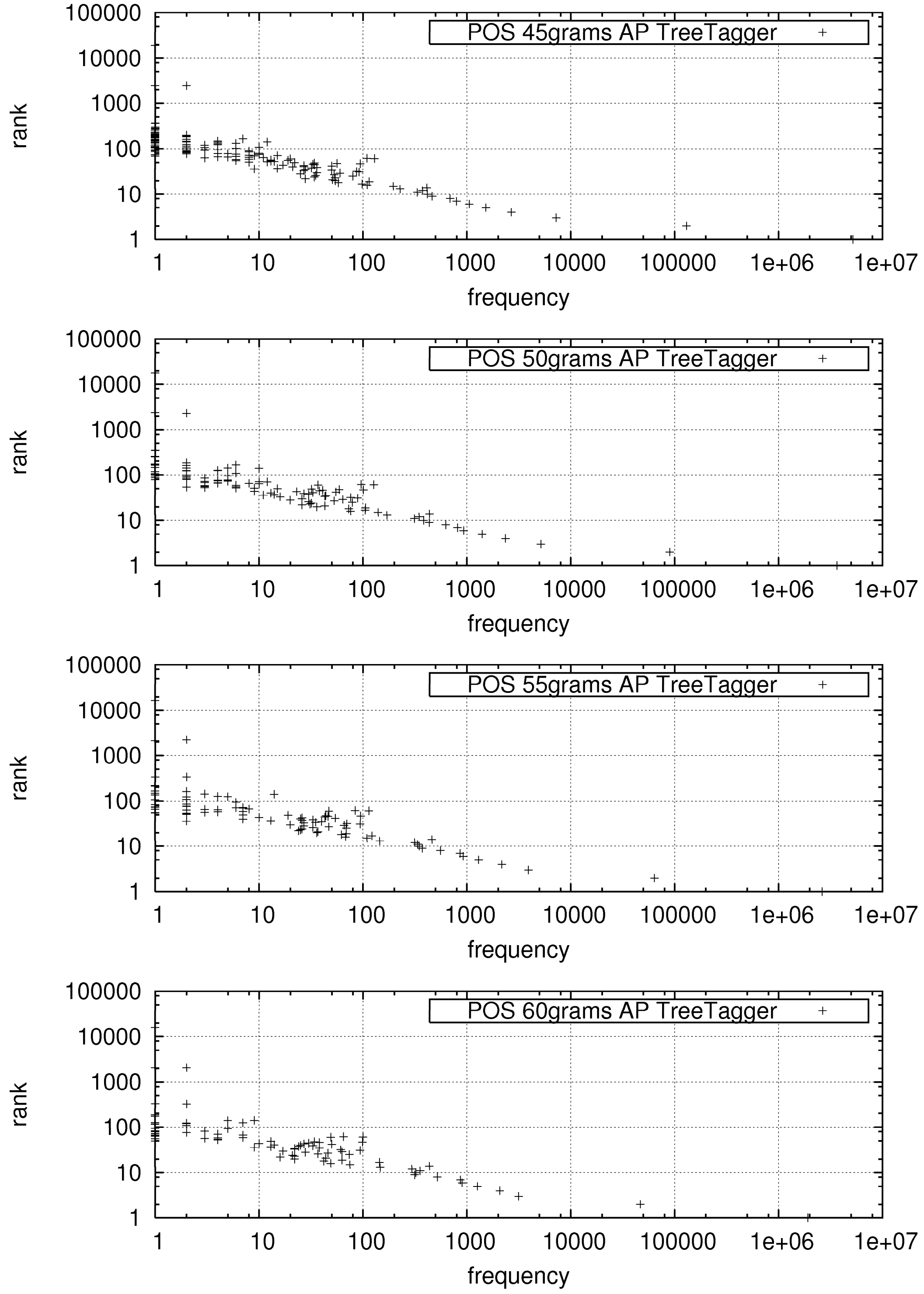


Figure B.8: Distribution of POS n-grams, for n = 45, 50, 55, 60 (AP).

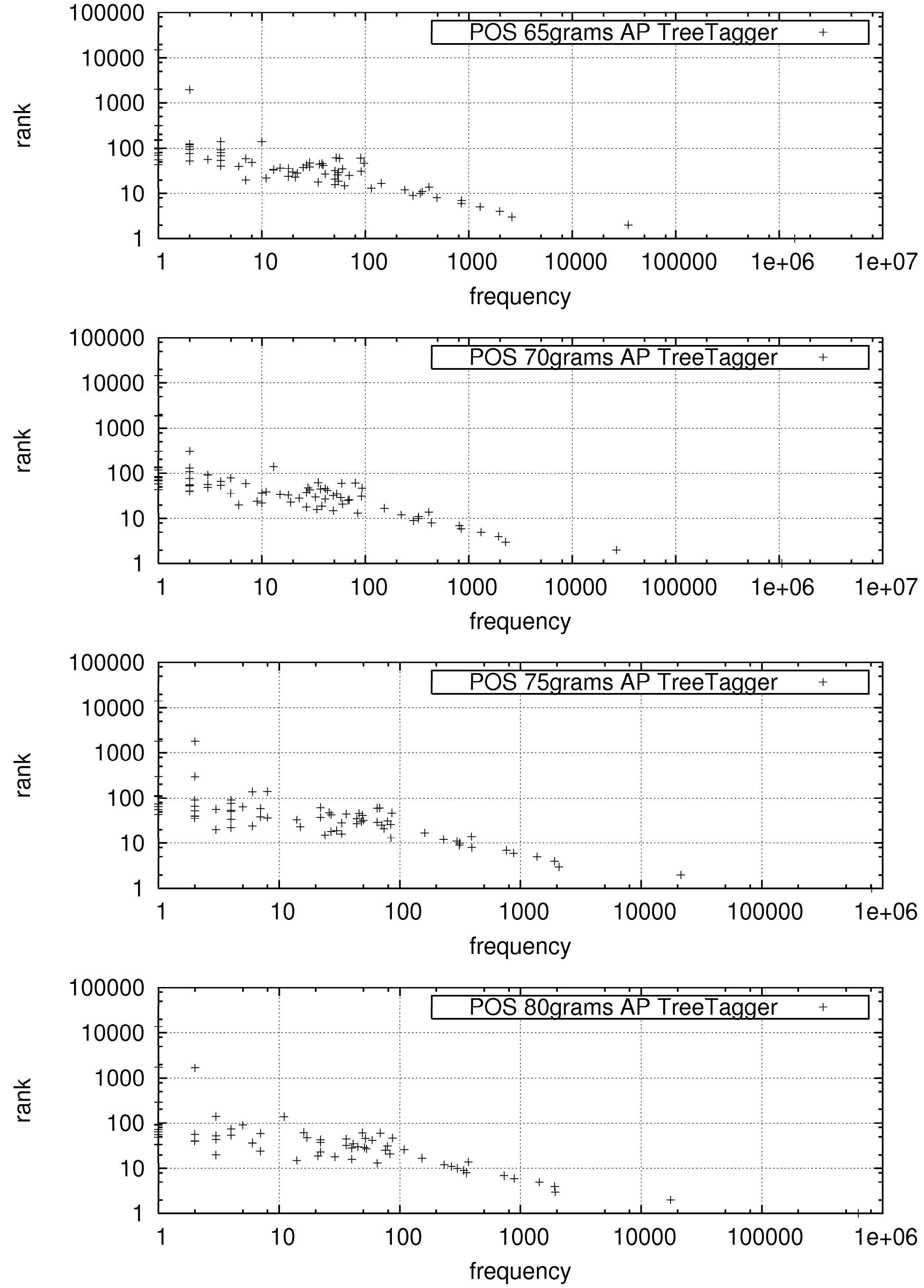


Figure B.9: Distribution of POS n-grams, for n = 65, 70, 75, 80 (AP).

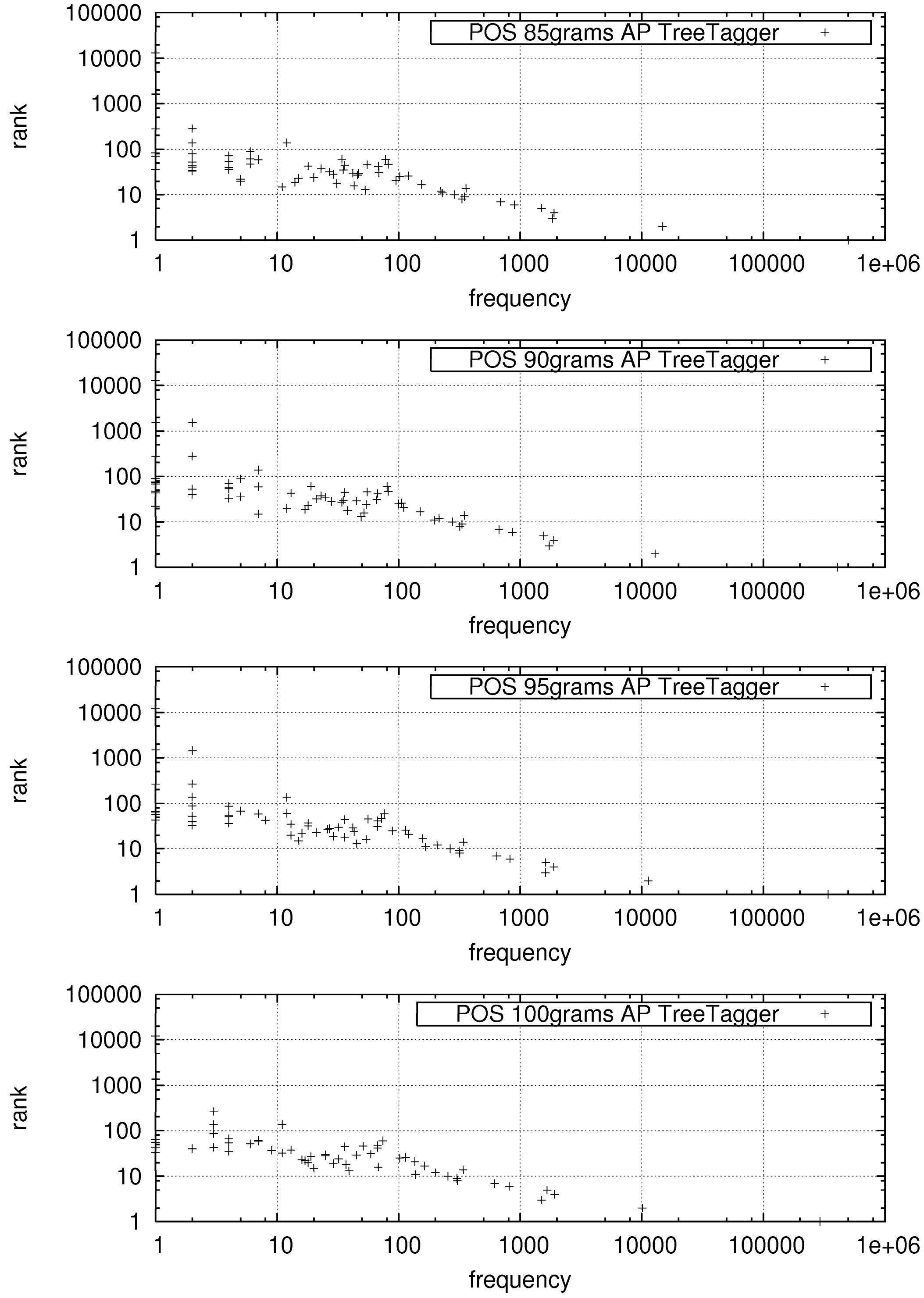


Figure B.10: Distribution of POS n-grams, for n = 85, 90, 95, 100 (AP).

Appendix C

Part of speech n-grams and informative content

Figures C.1 - C.3 relate to Chapter 7, and specifically Section ??, Figures 5.1 - 5.3, pages 60 - 61. Figures C.1 - C.3 plot POS n-gram frequency against the informative content of POS n-grams, for POS 5-grams, in five different collections. The corresponding plots for POS 4-grams are displayed in Figures 5.1 - 5.3, pages 60 - 61.

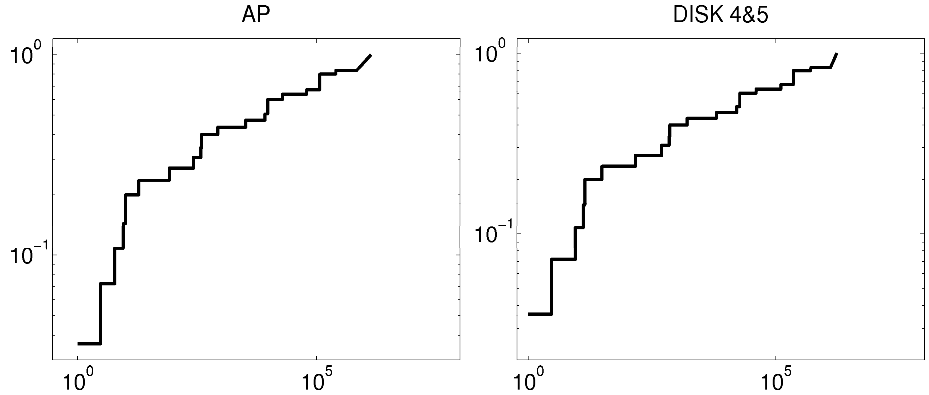


Figure C.1: Frequency versus informative content of POS 5-grams (AP, Disks 4&5).

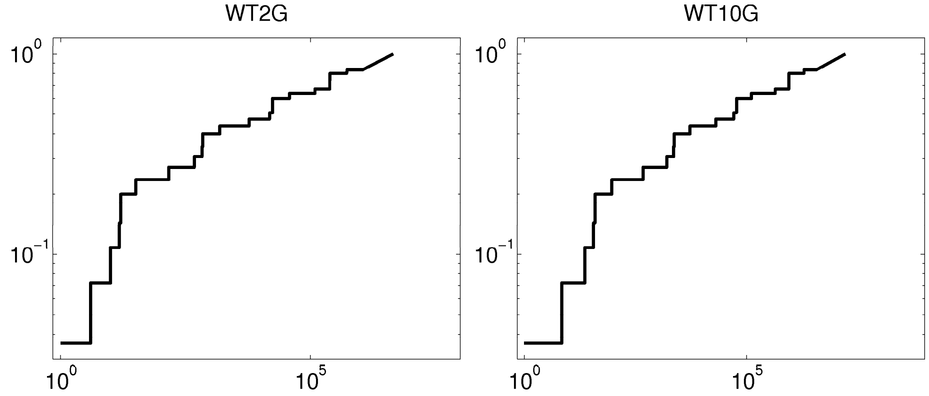


Figure C.2: Frequency versus informative content of POS 5-grams (WT2G, WT10G).

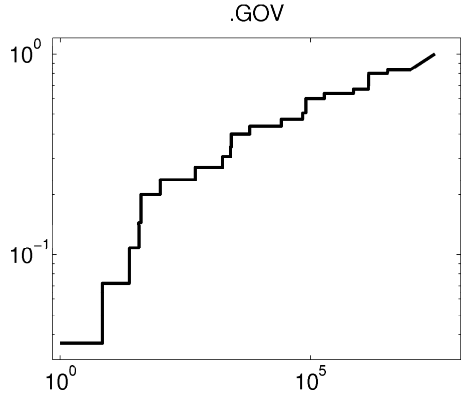


Figure C.3: Frequency versus informative content of POS 5-grams (.GOV collection).

Appendix D

Part of speech information score (PIS) and inverse document frequency (IDF)

Figures D.1 - D.17 relate to Chapter 6. Specifically, Figures D.1 - D.6 refer to Section 6.5.1, Table 6.10, page 92, and Figures D.7 - D.17 refer to Section 6.5.2, Table 6.11, page 95. The part of speech information score (PIS_2) plotted in these figures has been computed with Equation 5.21, page 73, using POS 4-grams from the collection specified in each caption. The collections, which have been presented in Section 6.3.2, Table 6.3, page 80, are POS tagged with the TreeTagger, which has been presented in Section 3.4.3, page 45.

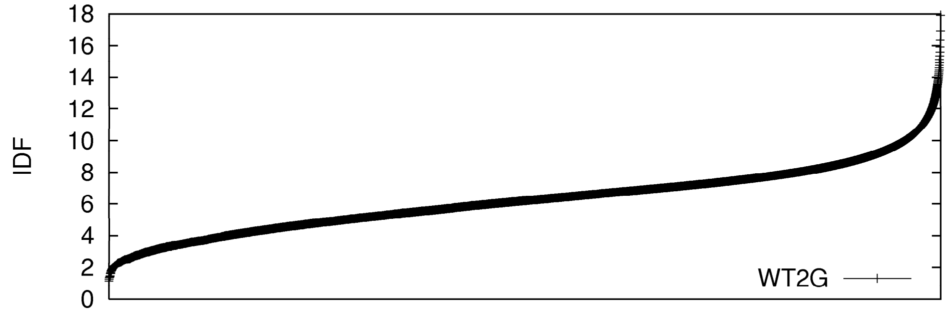


Figure D.1: IDF of terms in WT2G.

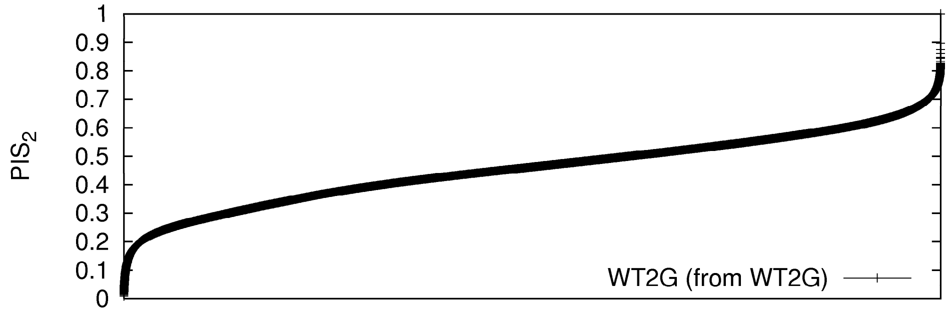


Figure D.2: PIS$_2$ of terms in WT2G (POS 4-grams from WT2G).

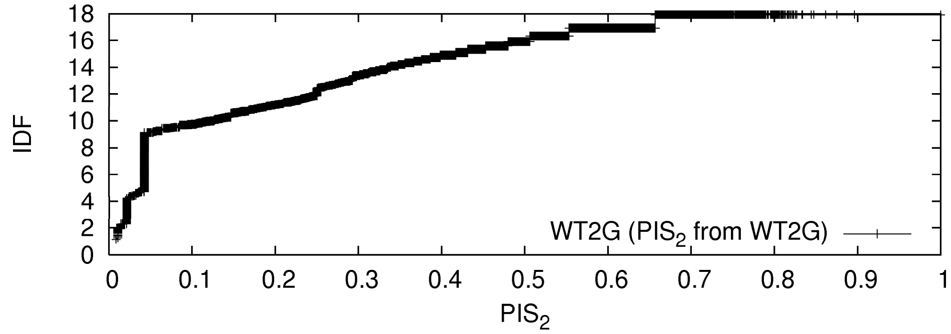


Figure D.3: PIS$_2$ versus IDF of terms in WT2G (POS 4-grams from WT2G).

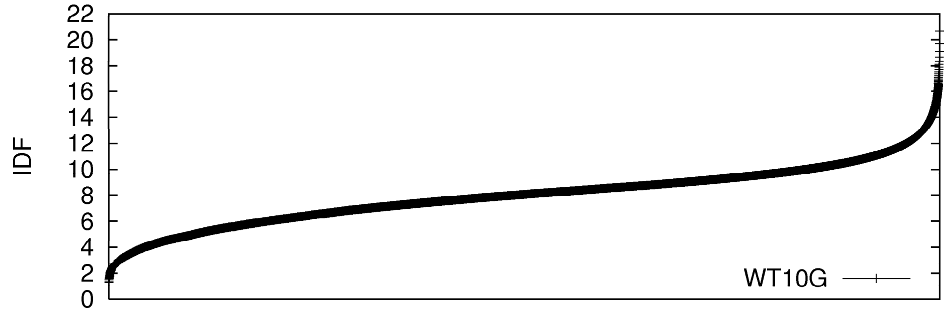


Figure D.4: IDF of terms in WT10G.

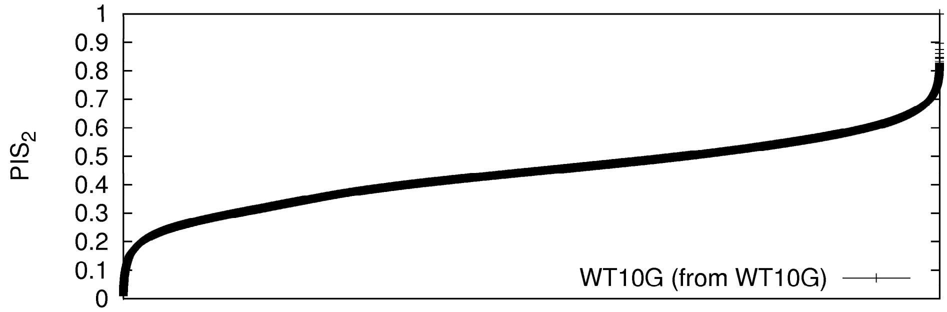


Figure D.5: PIS_2 of terms in WT10G (POS 4-grams from WT10G).

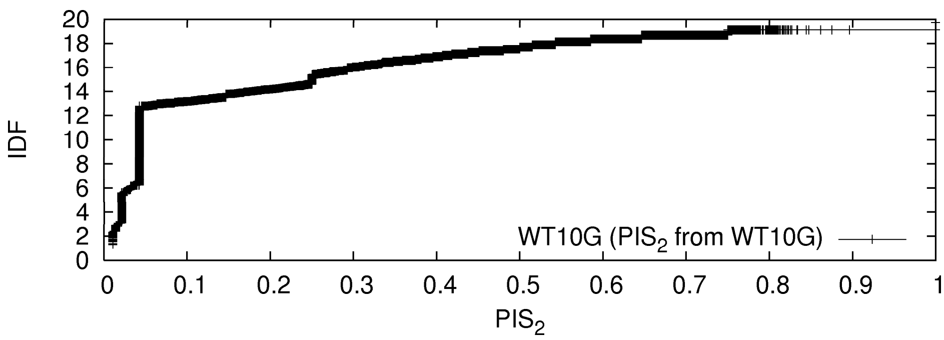


Figure D.6: PIS_2 versus IDF of terms in WT10G (POS 4-grams from WT10G).

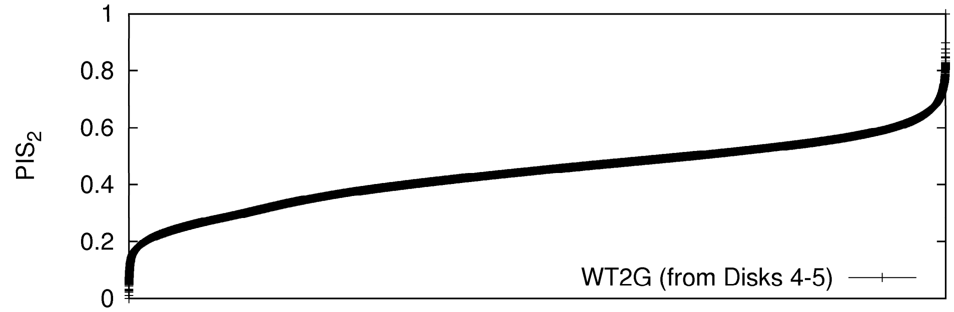


Figure D.7: PIS$_2$ of terms in WT2G (POS 4-grams from Disks 4&5).

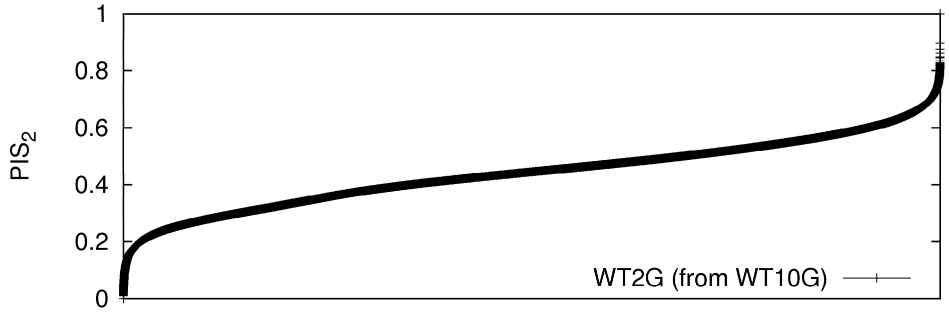


Figure D.8: PIS$_2$ of terms in WT2G (POS 4-grams from WT10G).

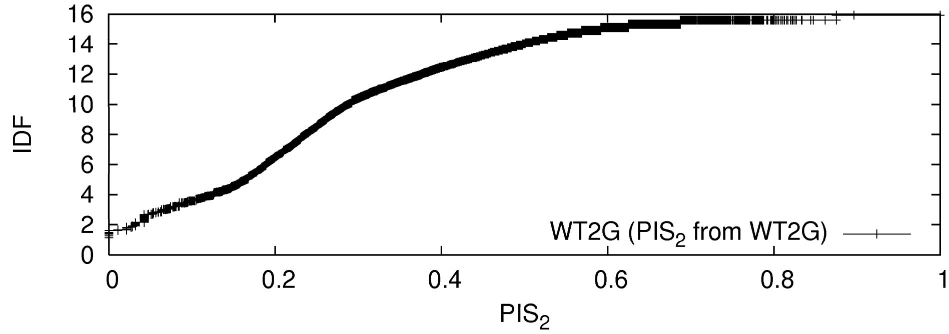


Figure D.9: PIS$_2$ versus IDF of terms in WT2G (POS 4-grams from Disks 4&5).

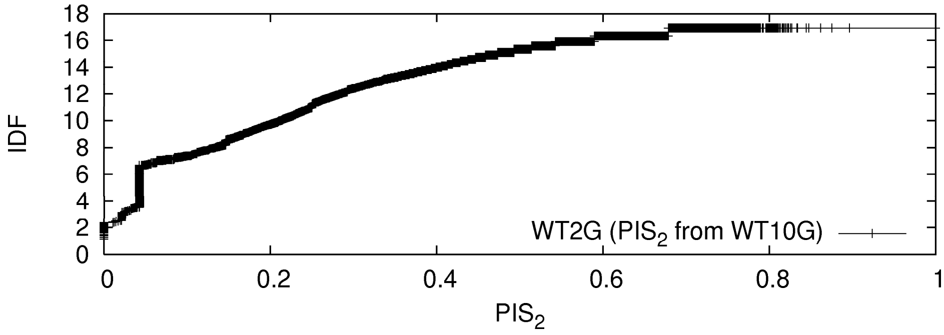


Figure D.10: PIS$_2$ versus IDF of terms in WT2G (POS 4-grams from WT10G).

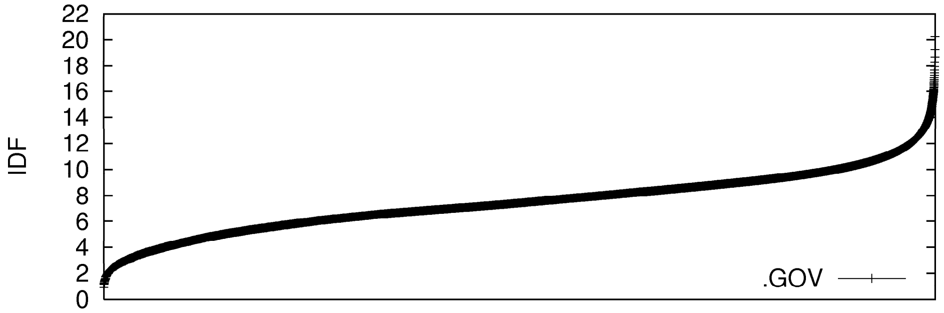


Figure D.11: IDF of terms in .GOV.

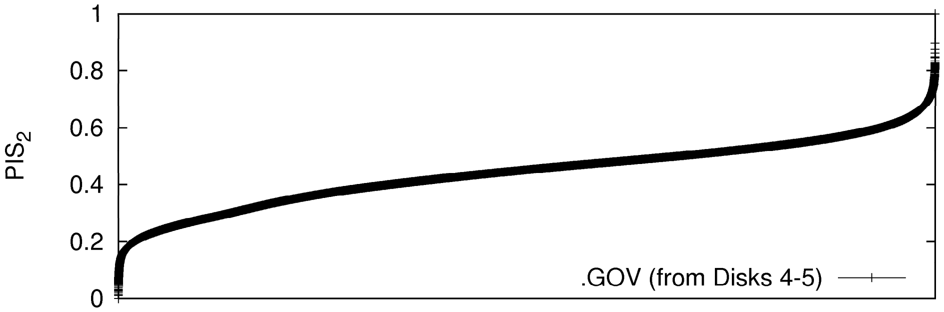


Figure D.12: PIS$_2$ of terms in .GOV (POS 4-grams from Disks 4&5 collection).

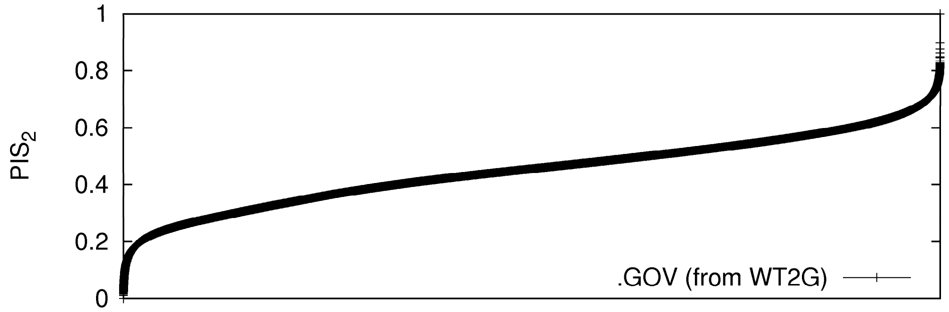


Figure D.13: PIS$_2$ of terms in .GOV (POS 4-grams from WT2G).

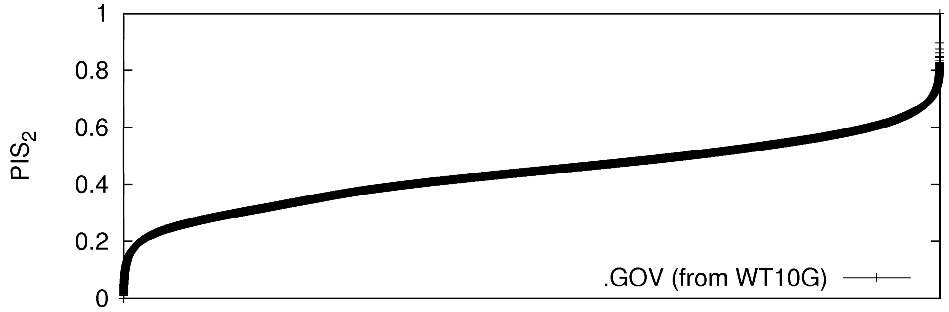


Figure D.14: PIS$_2$ of terms in .GOV (POS 4-grams from WT10G).

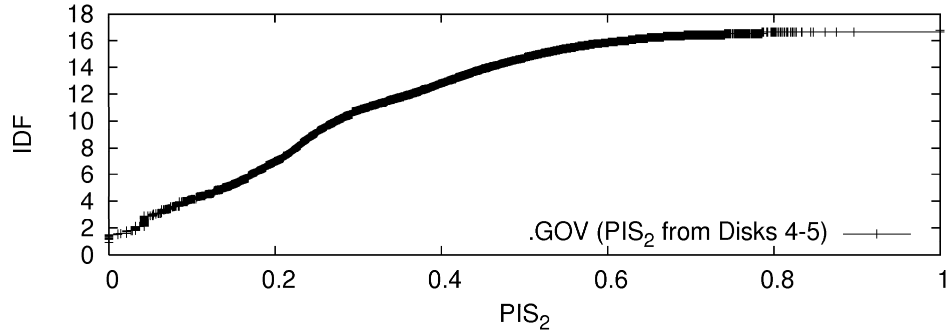


Figure D.15: PIS$_2$ versus IDF of terms in .GOV (POS 4-grams from Disks 4&5).

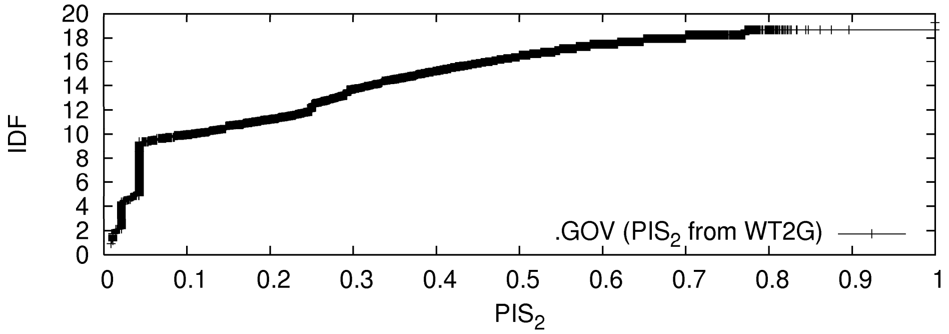


Figure D.16: PIS_2 versus IDF of terms in .GOV (POS 4-grams from WT2G).

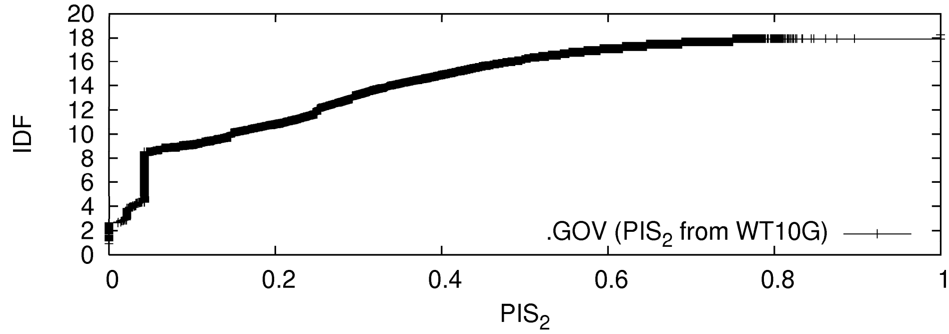


Figure D.17: PIS_2 versus IDF of terms in .GOV (POS 4-grams from WT10G).

Appendix E

Default and optimal parameter values

Tables E.1 and E.2 relate to Chapter 7, and specifically Section 7.3.3.5, Tables 7.11 and 7.12, pages 126 - 127. Table E.1 displays the default and optimal parameter values used in the experiments of Section 7.3.3.5, with short queries. Table E.2 displays the default and optimal parameter values used in the experiments of Section 7.3.3.5, with long queries.

Default and optimal parameters for short queries				
		WT10G		
model	setting	baseline	$f_{t,d} \times PIS_1$	$w_{t,d} \times PIS_1$
BM25	default MAP	b=0.75	b=0.75, λ=1.00, ϱ=0.30	b=0.75, λ=1.00, ϱ=0.30
	optimal MAP	b=0.25	b=0.25, λ=1.00, ϱ=0.90	b=0.25, λ=1.00, ϱ=0.30
BM25	default P10	b=0.75	b=0.75, λ=1.00, ϱ=0.30	b=0.75, λ=1.00, ϱ=0.30
	optimal P10	b=0.35	b=0.35, λ=1.00, ϱ=0.70	b=0.25, λ=1.00, ϱ=0.20
PL2	default MAP	c=7.00	c=7.00, λ=1.00, ϱ=0.90	c=7.00, λ=1.00, ϱ=0.30
	optimal MAP	c=12.0	c=16.0, λ=1.00, ϱ=0.90	c=16.0, λ=1.00, ϱ=0.20
PL2	default P10	c=7.00	c=7.00, λ=1.00, ϱ=0.30	c=7.00, λ=1.00, ϱ=0.30
	optimal P10	c=12.0	c=12.0, λ=1.00, ϱ=0.70	b=0.25, λ=1.00, ϱ=0.20
		Disks 4&5		
model	setting	baseline	$f_{t,d} \times PIS_1$	$w_{t,d} \times PIS_1$
BM25	default MAP	b=0.75	b=0.75, λ=1.00, ϱ=0.30	b=0.75, λ=1.00, ϱ=0.30
	optimal MAP	b=0.35	b=0.35, λ=1.00, ϱ=0.90	b=0.35, λ=1.00, ϱ=0.90
BM25	default P10	b=0.75	b=0.75, λ=1.00, ϱ=0.30	b=0.75, λ=1.00, ϱ=0.30
	optimal P10	b=0.35	b=0.35, λ=1.00, ϱ=0.90	b=0.35, λ=1.00, ϱ=0.50
PL2	default MAP	c=7.00	c=7.00, λ=1.00, ϱ=0.30	c=7.00, λ=1.00, ϱ=0.30
	optimal MAP	c=12.0	c=12.0, λ=1.00, ϱ=0.50	c=12.0, λ=1.00, ϱ=0.50
PL2	default P10	c=7.00	c=7.00, λ=1.00, ϱ=0.30	c=7.00, λ=1.00, ϱ=0.30
	optimal P10	c=12.0	c=12.0, λ=1.00, ϱ=0.50	c=12.0, λ=1.00, ϱ=0.50

Table E.1: Parameter values used in the experiments with short queries, reported in Section 7.3.3.5.

Default and optimal parameters for long queries				
		WT10G		
model	setting	baseline	$f_{t,d} \times PIS_1$	$w_{t,d} \times PIS_1$
BM25	default MAP	b=0.75	b=0.75, λ=1.00, ϱ=0.30	b=0.75, λ=1.00, ϱ=0.30
	optimal MAP	b=0.55	b=0.45, λ=1.00, ϱ=0.40	b=0.45, λ=1.00, ϱ=0.10
BM25	default P10	b=0.75	b=0.75, λ=1.00, ϱ=0.30	b=0.75, λ=1.00, ϱ=0.30
	optimal P10	b=0.55	b=0.25, λ=1.00, ϱ=0.30	b=0.25, λ=1.00, ϱ=0.20
PL2	default MAP	c=7.00	c=7.00, λ=1.00, ϱ=0.30	c=7.00, λ=1.00, ϱ=0.30
	optimal MAP	c=3.00	c=8.00, λ=1.00, ϱ=0.10	c=4.00, λ=1.00, ϱ=0.10
PL2	default P10	c=7.00	c=7.00, λ=1.00, ϱ=0.30	c=7.00, λ=1.00, ϱ=0.30
	optimal P10	c=2.00	c=6.00, λ=1.00, ϱ=0.10	c=8.00, λ=1.00, ϱ=0.20
model	setting	baseline	$f_{t,d} \times PIS_1$	$w_{t,d} \times PIS_1$
BM25	default MAP	b=0.75	b=0.75, λ=1.00, ϱ=0.30	b=0.75, λ=1.00, ϱ=0.30
	optimal MAP	b=0.65	b=0.55, λ=1.00, ϱ=0.10	b=0.55, λ=1.00, ϱ=0.20
BM25	default P10	b=0.75	b=0.75, λ=1.00, ϱ=0.30	b=0.75, λ=1.00, ϱ=0.30
	optimal P10	b=0.75	b=0.55, λ=1.00, ϱ=0.10	b=0.45, λ=1.00, ϱ=0.10
PL2	default MAP	c=7.00	c=7.00, λ=1.00, ϱ=0.30	c=7.00, λ=1.00, ϱ=0.30
	optimal MAP	c=2.00	c=6.00, λ=1.00, ϱ=0.30	c=4.00, λ=1.00, ϱ=0.10
PL2	default P10	b=0.75	b=0.75, λ=1.00, ϱ=0.30	b=0.75, λ=1.00, ϱ=0.30
	optimal P10	c=1.00	c=6.00, λ=1.00, ϱ=0.10	c=4.00, λ=1.00, ϱ=0.10

Table E.2: Parameter values used in the experiments with long queries, reported in Section 7.3.3.5.

Bibliography

AARONSON, S. (1999). Stylometric clustering. A comparative analysis of data-driven and syntactic features. Tech. rep., Berkeley University. 4.3.3

ADAMS, E. (1991). *A Study of Trigrams and their Feasibility as Index Terms in a Full Text Information Retrieval System*. PhD thesis, George Washington University. 4.2.2

ALDRICH, J. (1997). R. A. Fisher and the making of maximum likelihood 1912-1922. *Statistical Science*, **12**, 162–176. 2

AMATI, G. (2003). *Probabilistic Models for Information Retrieval based on Divergence from Randomness*. PhD thesis, University of Glasgow. 2.4.2.3, 2.4.2.3.2, 2.5.1, 7.2.2.3, 7.2.3.3, 7.3.3.4.3, 7.3.3.4.3, 7.3.3.4.4

AMATI, G., CARPINETO, C. & ROMANO, G., eds. (2007). *Proceedings of the 29th European Conference on Advances in Information Retrieval Research (ECIR 2007), Rome, Italy, 2007,*. E

ANDERSON, J.M. (1997). *A Notional Theory of Syntactic Categories*. Cambridge University Press. 3.3

ANGELL, R.C., FREUND, G.E. & WILLETT, P. (1983). Automatic spelling correction using trigram similarity measure. *Information Processing and Management*, **19**, 255–261. 4.2.2

ANICK, P.G. & TIPIRNENI, S. (1999). The paraphrase search assistant: terminological feedback for iterative information seeking. In Gey *et al.* (1999), 153–159. 3.5

Arampatzis, A., van Bommel, P., Koster, C. & van der Weide, T. (1997). Linguistic variation in information retrieval and filtering. Tech. rep., University of Nijmegen. 3.5

Argamon, A., Koppel, M. & Avneri, G. (1998a). Style-based text categorization: what newspaper am I reading? In M. Sahrami, ed., *Learning for Text Categorization: Papers from the American Association for Artificial Intelligence (AAAI) Workshop. Technical Report WS-98-05*, 1–4, California, USA. 4.2.2, 4.3.3

Argamon, S., Koppel, M. & Avneri, G. (1998b). Routing documents according to style. In D. Schwartz, M. Divitini & T. Brasethvik, eds., *Proceedings of the 1st International Workshop on Innovative Internet Information Systems (IIIS 1998)*, 1–13, Pisa, IT. 4.2.2, 4.3.3

Arusu, A., Cho, J., Garcia-Molina, H., Paepcke, A. & Raghavan, S. (2001). Searching the Web. *Transactions for Internet Technologies*, **1**, 2–43. 2.6.2

Aston, G. & Burnard, L. (1998). *The British National Corpus Handbook*. Edinburgh University Press. 4.3.3

Baayen, H., van Halteren, H. & Tweedie, F. (1996). Outside the cave of shadows: using syntactic annotation to enhance authorship attribution. *Literary and Linguistic Computing*, **11**, 121–131. 4.3.3

Baeza-Yates, R., Moffat, A. & Navarro, G. (2002). Searching large text collections. *Handbook of Massive Datasets*, 195–244. 2.3.2.1

Baeza-Yates, R.A. & Ribeiro-Neto, B.A. (1999). *Modern Information Retrieval*. ACM Press / Addison-Wesley. 2.1, 2.7

Baeza-Yates, R.A., Ziviani, N., Marchionini, G., Moffat, A. & Tait, J., eds. (2005). *Proceedings of the 28th Annual International Association for Computing Machinery (ACM) SIGIR Conference on Research and Development in Information Retrieval, Salvador, Brazil, 2005*. E

Bahl, L. & Jelinek, F. (1975). Decoding for channels with insertions, deletions and substitutions with application to speech recognition. *Transactions on Information Theory*, **21**, 404–411. 4.2.2

BAHL, L.R., JELINEK, F. & MERCER, R.L. (1990). A maximum likelihood approach to continuous speech recognition. *Readings in Speech Recognition*, 308–319. 4.2.2

BAKER, J.M. (1975). The DRAGON system - an overview. *Transactions on Acoustics, Speech, and Signal Processing*, **23**, 24–29. 4.2.2

BAS, A., DENISON, D., KEIZER, E. & POPOVA, G., eds. (2004). *Fuzzy Grammar, a Reader*. Oxford University Press. 3.2

BAXENDALE, P.B. (1958). Machine-made index for technical literature - an experiment. *IBM Journal for Research and Development*, **2**, 354–361. 2.5.2

BEELER, M., GOSPER, R.W. & SCHROEPPEL, R. (1972). Hakmem: Artificial intelligence memo no. 239. Tech. rep., Massachussetts Institute of Technology. 4.2.2

BELEW, R.K. (2000). *Finding Out About: A Cognitive Perspective on Search Engine Technology and the WWW*. Cambridge University Press. 2.1

BEST, K.H. (2001). Probability distributions of language entities. *Journal of Quantitative Linguistics*, **8**, 1–11. 5.3

BIRD, R.M., NEWSBAUM, J.B. & TREFFTZS, J.L. (1978). Text file inversion: an evaluation. In *Proceedings of the Workshop on Computer Architecture for Non-Numeric Processing*, 42–50. 1

BLAIR, D.C. (2002). Some thoughts on the reported results of TREC. *Information Processing Management*, **38**, 445–451. 2.7

BOITET, C. & WHITELOCK, P., eds. (1998). *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics (ACL) and 17th International Conference on Computational Linguistics (COLING), Montreal, Canada*. E

BOOKSTEIN, A. & SWANSON, D. (1974). Probabilistic models for automatic indexing. *Journal of the American Society of Information Science (JASIS)*, **25**, 312–318. 2.3.2.2

BRILL, E. (1995). Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics*, **21**, 543–565. 3.4.1, 6.3.1

Brown, P.F., Della Pietra, V.J., deSouza, P.V., Lai, J.C. & Mercer, R.L. (1992). Class-based n-gram models of natural language. *Computational Linguistics*, **18**, 467–479. 4.2.1, 4.2.2

Buckley, C. & Voorhees, E.M. (2004). Retrieval evaluation with incomplete information. In M. Sanderson, K. Järvelin, J. Allan & P. Bruza, eds., *SIGIR*, 25–32. 2.7

Buckley, C., Salton, G. & Allan, J. (1994). The effect of adding relevance information in a relevance feedback environment. In W.B. Croft & C.J.K. van Rijsbergen, eds., *SIGIR*, 292–300. 2.5.1

Burger, J.D., Palmer, D. & Hirschman, L. (1998). Named entity scoring for speech input. In Boitet & Whitelock (1998), 201–205. 4.2.2

Burnett, J.E., Cooper, D., Lynch, M.F., Willett, P. & Wycherley, M. (1979). Document retrieval experiments using indexing vocabularies of varying size. I. Variety generation symbols assigned to the fronts of index terms. *Journal of Documentation*, **35**, 197–206. 4.2.2

Canvar, W.B. (1993). N-gram-based text filtering for TREC-2. In D.K. Harman, ed., *Proceedings of the 2nd Text REtrieval Conference (TREC 1993)*, 171–180, NIST. 4.2.2

Canvar, W.B. (1994). Using an n-gram-based document representation with a vector processing retrieval model. In D.K. Harman, ed., *TREC*, 269–278, NIST. 4.2.2

Canvar, W.B. & Trenkle, J.M. (1994). N-gram-based text categorization. In *Proceedings of the 3rd Annual Symposium on Document Analysis and Information Retrieval, Las Vegas, USA*, 161–175. 4.2.2

Caraballo, S. & Charniak, E. (1999). Determining the specificity of nouns from text. In *Proceedings of the joint SIGDAT conference on Empirical Methods in Natural Language Processing (EMNLP) and Very Large Corpora (VLC)*, 63–70. 3.5

Carmel, D., Amitay, E., Herscovici, M., Maarek, Y.S., Petruschka, Y. & Soffer, A. (2001a). Juru at TREC 10 - experiments with index pruning. In Voorhees & Harman (2001), 228–236. 7.2.3.4

Carmel, D., Cohen, D., Fagin, R., Farchi, E., Herscovici, M., Maarek, Y.S. & Soffer, A. (2001b). Static index pruning for information retrieval systems. In Croft *et al.* (2001), 43–50. 7.2.3.4

Carmel, D., Yom-Tov, E., Darlow, A. & Pelleg, D. (2006). What makes a query difficult? In E.N. Efthimiadis, S.T. Dumais, D. Hawking & K. Järvelin, eds., *SIGIR*, 390–397. 2.5.1

Carpineto, C., de Mori, R., Romano, G. & Bigi, B. (2001). An information-theoretic approach to automatic query expansion. *Transactions for Information Systems*, **19**, 1–27. 2.5.1

Cheng, B.Y.M., G.Carbonell, J. & Klein-Seetharaman, J. (2005). Protein classification based on text document classification techniques. *Proteins: Structure, Function, and Bioinformatics*, **58**, 955–970. 4.2.2

Chiaramella, Y., ed. (1988). *Proceedings of the 11th Annual International Association for Computing Machinery (ACM) SIGIR Conference on Research and Development in Information Retrieval, Grenoble, France, 1988*. E

Chomsky, N. (1961). Some methodological remarks on generative grammar. *Word*, **17**, 219–239. 3.3

Chowdhury, A. & McCabe, M.C. (1998). Improving information retrieval systems using part of speech tagging. Tech. rep., University of Maryland. 3.5

Chowdhury, A., McCabe, M.C., Grossman, D.A. & Frieder, O. (2002). Document normalization revisited. In Jarvelin *et al.* (2002), 381–382. 7.3.3.4.4

Church, K.W. & Hanks, P. (1989). Word association norms, mutual information, and lexicography. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics (ACL), Vancouver, Canada*, 76–83. 3.4

Cohen, J. (1995). Highlights: Language- and domain-independent automatic indexing terms for abstracting. *Journal of the American Society for Information Science*, **46**, 162–174. 4.2.2

Cohen, P.R. & Wahlster, W., eds. (1997). *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL) and 8th Conference of the European Chapter of the Association for Computational Linguistics (EACL), Madrid, Spain*. E

COLLIER, R. (1994a). Empirical knowledge representation generation using n-gram clustering. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 1434. 4.2.2

COLLIER, R. (1994b). N-gram cluster identification during empirical knowledge representation generation. In *International Conference on Computational Linguistics (COLING)*, 1054–1058. 4.2.2

COOPER, W.S. & MARON, M.E. (1978). Foundations of probabilistic and utility-theoretic indexing. *Journal of the ACM*, **25**, 67–80. 2.3.2.2

CRASWELL, N., ROBERTSON, S.E., ZARAGOZA, H. & TAYLOR, M.J. (2005). Relevance weighting for query independent evidence. In Baeza-Yates *et al.* (2005), 416–423. 7.3.3.3.2

CRESTANI, F., SANDERSON, M. & THEOPHYLACTOU, M. (1997). Short Queries, Natural Language and Spoken Documents Retrieval: Experiments at Glasgow University. In Voorhees & Harman (1997), 667–686. 3.5

CROFT, B. & LAFFERTY, J. (2003). *Language Modeling for Information Retrieval*. Kluwer Academic Publishers. 4.2.2

CROFT, W.B., HARPER, D.J., KRAFT, D.H. & ZOBEL, J., eds. (2001). *Proceedings of the 24th Annual International Association for Computing Machinery (ACM) SIGIR Conference on Research and Development in Information Retrieval, New Orleans, USA, 2001*. E

CROWDER, G. & NICHOLAS, C. (1995). An approach to large scale distributed information systems using statistical properties of text to guide agent search. In *Proceedings of the Conferene on Information and Knowledge Management (CIKM) Workshop on Intelligent Information Agents*. 4.2.2

CRYSTAL, D. (1967). English word classes. *Lingua*, **7**, 24–56. 3.2

CUTTING, D.R., KUPIEC, J., PEDERSEN, J.O. & SIBUN, P. (1992). A practical part-of-speech tagger. In *Applied Natural Language Processing (ANLP) Conference*, 133–140. 4.2.2

DAMERAU, F.J. (1965). An experiment in automatic indexing. *American Documentation*, **16**, 283–289. 2.3.2.2

DAMERAU, F.J. (1971). *Markov Models and Linguistic Theory*. Mouton. 4.2.1

DEMARTINI, G. & MIZZARO, S. (2006). A classification of ir effectiveness metrics. In M. Lalmas, A. MacFarlane, S.M. Rüger, A. Tombros, T. Tsikrika & A. Yavlinsky, eds., *ECIR*, vol. 3936 of *Lecture Notes in Computer Science*, 488–491, Springer. 2.7

DEROSE, S.J. (1988). Grammatical category disambiguation by statistical optimization. *Computational Linguistics*, **14**, 31–39. 3.4

DILLON, M. & GRAY, A. (1983). FASIT - A fully automatic syntactically based indexing system. *Journal of the American Society of Information Science*, **34**, 99–108. 2.5.2, 3.5

DOYLE, L.B. (1962). Indexing and abstracting by association. *American Documentation*, **13**, 378–390. 2.5.2

EARL, L.L. (1972). The resolution of syntactic ambiguity in automatic language processing. *Information Storage and Retrieval*, **8**, 277–308. 2.5.2

EVANS, D.A. & ZHAI, C. (1996). Noun-phrase analysis in unrestricted text for information retrieval. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics (ACL), Santa Cruz, USA*, 17–24. 3.5

FAGAN, J.L. (1987). Automatic phrase indexing for document retrieval: An examination of syntactic and non-syntactic methods. In SIGIR 1987, 91–101. 3.5

FAGAN, J.L. (1989). The effectiveness of a nonsyntactic approach to automatic phrase indexing for document retrieval. *Journal of the American Society of Information Science*, **40**, 115–132. 2.5.2

FELLER, W. (1950). *An Introduction to the Probability Theory and its Application*. Wiley. 4.2.2

FENG, L., UMEMURA, K., YAMAMOTO, M. & CHURCH, K. (2000). Using variable length ngrams for retrieving technical abstracts in Japanese. In *Information Retrieval with Asian Languages IRAL*, 213–214. 4.2.2

FLANK, S. (1998). A layered approach to NLP-based information retrieval. In Boitet & Whitelock (1998), 397–403. 3.5

Fox, E., Betrabet, S., Koushik, M. & Lee, W. (1992). Extended boolean models. In Frakes & Baeza-Yates (1992), 393–419. 2.4.2.1

Frakes & Baeza-Yates, eds. (1992). *Information Retrieval: Data Structures and Algorithms*. Englewood Cliffs: Prentice Hall. W.B. 2.6, E

Francis, W.N. & Kuĉera, H. (1982). *Frequency Analysis of English Usage*. Houghton Mifflin. 5.2

Frei, H.P., Harman, D., Schäuble, P. & Wilkinson, R., eds. (1996). *Proceedings of the 19th Annual International Association for Computing Machinery (ACM) SIGIR Conference on Research and Development in Information Retrieval, Zurich, Switzerland, 1996*. E

Fuhr, N. (1989). Models for retrieval with probabilistic indexing. *Information Processing and Management*, **25**, 55–72. 2.3.2.2

Fujita, S. (2001). More reflections on "aboutness" TREC-2001 evaluation experiments at Justsystem. In Voorhees & Harman (2001), 331–338. 3.5, 7.2.2.4

Gallager, R.G. (1968). *Information Theory and Reliable Communication*. Wiley. 4.2.2

Garside, R., Leech, G. & Sampson, G. (1987). *The Computational Analysis of English: A Corpus-Based Approach*. Longman. 3.4

Gey, F., Hearst, M. & Tong, R., eds. (1999). *Proceedings of the 22nd Annual International Association for Computing Machinery (ACM) SIGIR Conference on Research and Development in Information Retrieval, Berkeley, USA, 1999*. E

Giuliano, V.E. & Jones, P.E. (1963). Linear associative information retrieval. *Vistas in Information Handling: The Augmentation of Man's Intellect by Machine*, **1**, 30–54. 2.5.2

Good, I.J. (1968). *The Estimation of Probabilities: an Essay of Modern Bayesian Methods*. MIT Press. 5.3

Grossman, D.A. & Frieder, O. (2004). *Information Retrieval: Algorithms and Heuristics*. Kluwer Academic Publishers. 2.6

HARDING, S.M., CROFT, W.B. & WEIR, C. (1997). Probabilistic retrieval of OCR degraded text using n-grams. In C. Peters & C. Thanos, eds., *Proceedings of the 1st European Conference on Research and Advanced Technology for Digital Libraries (ECDL 1997), Pisa, Italy, 1-3 September,*, 345–359. 4.2.2

HARMAN, D. (1987). A failure analysis on the limitations of suffixing in an online environment. In SIGIR 1987, 102–108. 2.3.1.2

HARMAN, D. (1988). Towards interactive query expansion. In Chiaramella (1988), 321–331. 2.5.1

HARMAN, D.K. (1991a). How effective is suffixing? *Journal of the American Society for Information Science*, **42**, 7–15. 2.3.1.2

HARMAN, D.K. (1991b). Ranking algortihms. *Information Retrieval: Data Structures and Algorithms*, 362–392. 2.3.1.2

HARPER, D.J. & VAN RIJSBERGEN, C.J.K. (1978). An evaluation of feedback in document retrieval using cooccurrence data. *Journal of Documentation*, **34**, 189–216. 2.5.2

HARTER, S.P. (1974). *A Probabilistic Approach to Automatic Keyword Indexing*. PhD thesis, Graduate Library, the University of Chicago, Chicago. 2.3.2.2

HASKIN, R.L. (1980). Hardware for searching very large databases. In *Proceedings of the Workshop on Computer Architecture for Non-Numeric Processing, Pacific Grove, CA, USA*, 49–56. 1

HE, B. & OUNIS, I. (2003). A study of parameter tuning for term frequency normalization. In *Proceedings of the 12th international Conference on Information and Knowledge Management (CIKM), New Orleans, USA*, 10–16. 7.3.3.4.4

HE, B. & OUNIS, I. (2005a). A study of the dirichlet priors for term frequency normalisation. In Baeza-Yates *et al.* (2005), 465–471. 7.3.3.4.4

HE, B. & OUNIS, I. (2005b). Term frequency normalisation tuning for BM25 and DFR models. In Losada & Fernández-Luna (2005), 200–214. 7.3.3.4.4

HEYLIGHEN, F. & DEWAELE, J.M. (2002). Variation in the contextuality of language: An empirical measure. *Foundations of Science*, **7**, 293–340. 1

HIEMSTRA, D. (2000). A probabilistic justification for using tf x idf term weighting in information retrieval. *International Journal on Digital Libraries*, **3**, 131–139. 2.4.2.4

HIEMSTRA, D. (2001). *Using Language Models for Information Retrieval*. Ph.D. thesis, University of Twente. 4.2.2

HJELMSLEV, L. (1943). *Prolegomena to a Theory of Language (Translated from Danish by F. J Whitfield)*. Bloomington. 3.2

HOPPER, P. & THOMPSON, S. (1984). The discourse basis for lexical categories in universal grammar. *Language*, **60**, 703–752. 3.2

HUDSON, R. (1994). About 37% of word-tokens are nouns. *Language*, **70**, 331–339. 5.2

HUFFMAN, S. & DAMASHEK, M. (1994). Acquaintance: A novel vector-space n-gram technique for document categorization. In E.M. Voorhees & L.P. Buckland, eds., *TREC*, 305–310, NIST. 4.2.2

HULL, D.A. (1996). Stemming algorithms: a case study for detailed evaluation. *Journal of the American Society for Information Science*, **47**, 70–84. 2.3.1.2

HULL, J.J. & SRIHARI, S.N. (1982). Experiments in text recognition with binary n-gram and viterbi algorithms. *Institute of Electrical and Electronics Engineers (IEEE) Transactions on Pattern Analysis and Machine Intelligence*, **4**, 520–530. 4.2.2

IVIE, E.L. (1966). *Search Procedure based on Measures of Relatedness between Documents*. Ph.D. thesis, Massachussetts Institute of Technology. 2.4.2.2

JACOBS, P.S. (1992). Joining statistics with NLP for text categorization. In *Applied Natural Language Processing (ANLP)*, 178–185. 3.5

JACOBS, P.S. & RAU, L.F. (1993). Innovations in text interpretation. *Artificial Intelligence*, **63**, 143–191. 3.5

JACQUEMIN, C., KLAVANS, J.L. & TZOUKERMANN, E. (1997). Expansion of multi-word terms for indexing and retrieval using morphology and syntax. In Cohen & Wahlster (1997), 24–31. 3.5

JARVELIN, K., BEAULIEU, M., BAEZA-YATES, R. & MYAENG, S.H., eds. (2002). *Proceedings of the 25th Annual International Association for Computing Machinery (ACM) SIGIR Conference on Research and Development in Information Retrieval, Tampere, Finland, 2002*. E

JELINEK, F. (1977). Continuous speech recognition. *SIGART Bulletin*, **61**, 33–34. 4.2.2

JESPERSEN, O. (1913). *Sprogets Logik (The Logic of Language)*. University of Copenhagen. 1.5, 3.3, 5.3.4.3

JESPERSEN, O. (1929). *The Philosophy of Grammar*. Allen and Unwin. 1.5, 3.3, 5.3.4.3

JOHANNES, F. (1998). A study using n-gram features for text categorization. Tech. rep., Austrian Institute for Artificial Intelligence. 4.2.2, 4.3.3

JURAFSKY, D. & MARTIN, J.H. (2000). *Speech and Language Processing: an Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall PTR. 2

KANG, B.Y. & LEE, S.J. (2005). Document indexing: a concept-based approach to term weight estimation. *Information Processing and Management*, **41**, 1065–1080. 2.3.2.2

KARLGREN, J. (1993). Syntax in information retrieval. In *Proceedings of the 1st Nordic Doctoral Symposium on Computational Linguistics, Copenhagen, Denmark*. 3.5

KESSLER, B., NUNBERG, G. & SCHÜTZE, H. (1997). Automatic detection of text genre. In Cohen & Wahlster (1997), 32–38. 4.3.3

KILGARRIFF, A. (2005). Language is never, ever, ever, random. *Corpus Linguistics and Linguistic Theory*, **1**, 263–275. 2.5.2

KIM, J.Y. & SHAWE-TAYLOR, J. (1994). Fast string matching using an n-gram algorithm. *Software-Practice and Experience*, **24**, 79–88. 4.2.2

KOBAYASHI, M. & TAKEDA, K. (2000). Information retrieval on the Web. *Computer Survey*, **32**, 144–173. 2.3.1, 2.6.2

Koppel, M., Akiva, N. & Dagan, I. (2003a). A corpus-independent feature set for style-based text categorization. In *International Joint Conference on Artificial Intelligence (IJCAI) Workshop on Computational Approaches to Text Style and Synthesis, Acapulco, Mexico*. 4.3.3

Koppel, M., Argamon, S. & Shimoni, A.R. (2003b). Automatically categorizing written texts by author gender. *Literary and Linguistic Computing*, 401–412. 4.3.3

Kraaij, W. (2004). *Variations on Language Modeling for Information Retrieval*. Ph.D. thesis, University of Twente. 4.2.2

Kraaij, W., Westerveld, T. & Hiemstra, D. (2002). The importance of prior probabilities for entry page search. In Jarvelin *et al.* (2002), 27–34. 7.3.3.3.2

Krovetz, R. (1993). Viewing morphology as an inference process. In R. Korfhage, E.M. Rasmussen & P. Willett, eds., *SIGIR*, 191–202. 2.3.1.2

Kurai, R., Minato, S.I. & Zeugmann, T. (2006). N-gram analysis based on zero-suppressed BDDs. In T. Washio, K. Satoh, H. Takeda & A. Inokuchi, eds., *JSAI*, 289–300. 4.2.2

Lafferty, J. & Zhai, C. (2003). *Probabilistic relevance models based on document and query generation*. Kluwer. 2.4.2, 2.4.2.4

Lafferty, J.D. & Zhai, C. (2001). Document language models, query models, and risk minimization for information retrieval. In Croft *et al.* (2001), 111–119. 2.4.2.4

Lancaster, F.W. & Fayen, E.G. (1973). *Information Retrieval Online*. Melville. 2.1

Lavrenko, V. & Croft, W.B. (2001). Relevance-based language models. In Croft *et al.* (2001), 120–127. 2.4.2.4

Lesk, M.E. (1969). Word-word associations in document retrieval systems. *American Documentation*, **20**, 27–38. 2.5.2

Lewis, D.D. (1992). An evaluation of phrasal and clustered representations on a text categorization task. In N.J. Belkin, P. Ingwersen & A.M. Pejtersen, eds., *SIGIR*, 37–50. 2.5.2

Lewis, D.D. & Croft, W.B. (1990). Term clustering of syntactic phrases. In J.L. Vidick, ed., *SIGIR'90, 13th International Conference on Research and Development in Information Retrieval, Brussels, Belgium, 5-7 September 1990, Proceedings*, 385–404, ACM Press. 3.5

Li, W. (1992). Random texts exhibit Zipf's-law-like word frequency distribution. *Transactions on Information Theory*, **38**, 1842–1845. 6.3.2.2

Lim, C.S., Lee, K.J. & Kim, G.C. (2005). Multiple sets of features for automatic genre classification of Web documents. *Information Processing and Management*, **41**, 1263–1276. 4.3.3

Lin, D. (1995). Description of the PIE System as used for MUC-6. In *Proceedings of the 6th Conference on Message Understanding (MUC-6)*, 67–81. 3.5

Lin, J. (2001). *Indexing and Retrieving Natural Language Using Ternanry Expressions*. Master thesis, Massachussetts Institute of Technology. 3.5

Lioma, C. & Ounis, I. (2005). Deploying part-of-speech patterns to enhance statistical phrase-based machine translation resources. In *Proceedings of the ACL Workshop on Building and Using Parallel Texts*, 163–166, Association for Computational Linguistics, Ann Arbor, Michigan. 1.6

Lioma, C. & Ounis, I. (2006). Examining the content load of part of speech blocks for information retrieval. In *ACL*, The Association for Computer Linguistics. 1.6

Lioma, C. & Ounis, I. (2007a). Extending weighting models with a term quality measure. In N. Ziviani & R.A. Baeza-Yates, eds., *SPIRE*, vol. 4726 of *Lecture Notes in Computer Science*, 205–216, Springer. 1.6

Lioma, C. & Ounis, I. (2007b). Light syntactically-based index pruning for information retrieval. In G. Amati, C. Carpineto & G. Romano, eds., *ECIR*, vol. 4425 of *Lecture Notes in Computer Science*, 88–100, Springer. 1.6

Lioma, C. & Ounis, I. (2008). A syntactically-based query reformulation technique for information retrieval. *Information Processing and Management*, **44**, 143–162. 1.6

Lioma, C. & van Rijsbergen, C.J.K. (2008). Part of speech n-grams and information retrieval. *Revue Française de Linguistique Appliquèe.*, **XIII**, 9–22. 1.6

LIOMA, C., MACDONALD, C., PLACHOURAS, V., PENG, J. & HE, I., B. AND'OUNIS (2006). University of glasgow at trec 2006: Experiments in terabyte and enterprise tracks with terrier. In *In Proceedings of the Text REtrieval Conference (TREC 2006)*, National Institute of Standards and Technology. 1.6

LOSADA, D.E. & FERNÁNDEZ-LUNA, J.M., eds. (2005). *Proceedings of the 27th European Conference on Advances in Information Retrieval Research (ECIR 2005), Santiago de Compostela, Spain, 2005*. E

LOSEE, J.R. (1994). Term dependence: truncating the Bahadur Lazarsfeld expansion. *Information Processing and Management*, **30**, 293–303. 2.5.2

LUHN, H.P. (1958). The automatic creation of literature abstracts. *IBM Journal of Research and Development*, **2**, 159–165. 2.3.2.2

LUHN, H.P. (1960). Keyword-in-context index for technical literature. *American Documentation*, **11**, 288–295. 2.3.2.2

LYONS, J. (1968). *Introduction to Theoretical Linguistics*. Cambridge University Press. 3

LYONS, J. (1977). *Semantics: Volume 2*. Cambridge University Press. 3.2, 3.2

MANI, I. (2001). *Automatic Summarization*. John Benjamins Publishing Company. 3.2, 3.5

MANNING, C.D. & SCHUTZE, H. (1999). *Foundations of Statistical Language Processing*. The MIT Press. 2

MARCHIONINI, G. (1995). *Information Seeking in Electronic Environments*. Cambridge University Press. 2.2

MARCUS, M.P., SANTORINI, B. & MARCINKIEWICZ, M.A. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, **19**, 313–330. 3.2, 3.4, 6.3.1, A

MARIÒO, J.B., BANCHS, R.E., CREGO, J.M., DE GISPERT, A., LAMBERT, P., FONOLLOSA, J.A.R. & COSTA-JUSSÀ, M.R. (2006). N-gram-based machine translation. *Computational Linguistics*, **32**, 527–549. 4.2.2

MARKOV, A.A. (1913). Essai d'une recherche statistique sur le texte du roman "Eugene Onegin" illustrant la liaison des epreuve en chain ('example of a statistical investigation of the text of "Eugene Onegin" illustrating the dependence between samples in chain'). English translation by Morris Halle, 1956. *Izvistia Imperatorskoi Akademii Nauk (Bulletin de l'Academie Imperiale des Sciences de St.-Petersbourg)*, **7**, 153–162. 4.2.1, 4.2.2

MARON, M.E. & KUHNS, J.L. (1960). On relevance, probabilistic indexing, and information retrieval. *Journal of the ACM*, **7**, 216–244. 2.3.2.2, 2.4.2.3

MCELWAIN, C.K. & EVENS, M.B. (1962). The Degarbler – A program for correcting machine-read Morse code. *Information and Control*, **5**, 368–384. 4.2.2

MEHMET, F. (1990). *Optimizing a Text Retrieval System Utilizing N-gram Indexing*. PhD thesis, George Washington University. 4.2.2

METZLER, D. (2006). Estimation, sensitivity, and generalization in parameterized retrieval models. In P.S. Yu, V.J. Tsotras, E.A. Fox & B. Liu, eds., *Conference on Information and Knowledge Management (CIKM)*, 812–813, ACM Press. 7.3.3.4.5

METZLER, D. & CROFT, W.B. (2005). A markov random field model for term dependencies. In Baeza-Yates *et al.* (2005), 214–221. 2.5.2

METZLER, D.P., NOREAULT, T., RICHEY, L. & HEIDORN, B. (1984). Dependency parsing for information retrieval. In C.J.K. van Rijsbergen, ed., *SIGIR*, 313–324. 2.5.2

MIHALCEA, R. (2003). Performance analysis of a part of speech tagging task. In A.F. Gelbukh, ed., *Proceedings of the 4th International Conference on Computational Linguistics and Intelligent Text Processing (CICLing 2003), Mexico City, Mexico, February 16-22, 2003*, 158–167. 3.4, 3.4.1, 3.4.2, 3.4.3, 6.3.1, 6.3.1

MIKK, J. (2000). *Textbook: Research and Writing*. Frankfurt am Main etc.: Peter Lang. 3.5

MIKK, J. (2001). Prior knowledge of text content and values of text characteristics. *Journal of Quantitative Linguistics*, **8**, 67–80. 3.5, 8.2.5

MILLER, D.R.H., LEEK, T. & SCHWARTZ, R.M. (1999). A hidden markov model information retrieval system. In Gey *et al.* (1999), 214–221. 2.4.2.4

MILLER, G.A. (1951). *Language and Communication*. McGraw-Hill, New York. 5.2

MISHNE, G. & DE RIJKE, M. (2005). Boosting web retieval through query operations. In Losada & Fernández-Luna (2005), 502–516. 2.5.2

MITZENMACHER, M. (2004). A brief history of generative models for power law and lognormal distributions. *Internet Mathematics*, **1**, 226–251. 6.3.2.2

NALLAPATI, R. & ALLAN, J. (2002). Capturing term dependencies using a language model based on sentence trees. In *Conference on Information and Knowledge Management (CIKM)*, 383–390, ACM Press. 2.5.2

NARANAN, S. & BALASUBRAHMANYAN, V. (1998). Models for power law relations in linguistics and in information science. *Journal of Quantitative Linguistics*, **5**, 35–61. 6.3.2.2

NARITA, M. & OGAWA, Y. (2000). The use of phrases from query texts in information retrieval. In N.J. Belkin, P. Ingwersen & M.K. Leong, eds., *SIGIR*, 318–320. 3.5

NEWMAN, M. (2005). Power laws, Pareto distributions and Zipf's law. *Contemporary Physics*, **46**, 323–351. 6.3.2.2

NEWMEYER, F. (2000). The discrete nature of syntactic categories: Against a prototype-based account. *The Nature and Foundation of Syntactic Categories, Syntax and Semantics*, **32**, 221–250. 3.3

NG, K. (1999). A maximum likelihood ratio information retrieval model. In E.M. Voorhees & D.K. Harman, eds., *TREC*, 267–274, NIST. 2.4.2.4

OCH, F.J. & NEY, H. (2004). The alignment template approach to statistical machine translation. *Computational Linguistics*, **30**, 417–449. 4.2.2

O'GRADY, W. (1988). Principles of grammar and learning. *Language*, **64**, 167–171. 3.3

OUNIS, I., LIOMA, C., MACDONALD, C. & PLACHOURAS, V. (2007). Research directions in terrier. *Novatica/UPGRADE Special Issue on Web Information Access, Ricardo Baeza-Yates et al. (Eds), Invited Paper*. 1.6, 6.4, 7.3.2.3.2

OZMUTLU, S., SPINK, A. & OZMUTLU, H.C. (2004). A day in the life of Web searching: an exploratory study. *Information Processing and Management*, **40**, 319–345. 5.2, 7.2.3.3

PEDERSON, J., SILVERSTEIN, C. & VOGT, C. (1997). Verity at TREC-6: Out of the box and beyond. In Voorhees & Harman (1997), 259–274. 3.5

PLACHOURAS, V. & OUNIS, I. (2007). Multinomial randomness models for retrieval with document fields. In Amati *et al.* (2007), 28–39. 2.5.2

PONTE, J.M. & CROFT, W.B. (1998). A language modeling approach to information retrieval. In *SIGIR*, 275–281. 2.4.2.4

PORTER, M.F. (1980). An algorithm for suffix stripping. *Program*, **14**, 130–137. 2.3.1.2, 6.4, 7.3.2.3.2

QUINLAN, J.R. (1983). Inferno: A cautious approach to uncertain inference. *Computer Journal*, **26**, 255–269. 3.4.3

RABINER, L.R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the Institute of Electrical and Electronics Engineers (IEEE)*, **77**, 257–286. 4.2.2

RADFORD, A. (1988). *Transformational Grammar*. Cambridge University Press. 3.2

RATNAPARKHI, A. (1996). A maximum entropy model for part-of-speech tagging. In *Proceedings of the Empirical Methods in Natural Language Processing Conference*, 130–142. 3.4.2, 4.2.2, 6.3.1

RATNAPARKHI, A. (2000). Trainable methods for surface natural language generation. In *Applied Natural Language Processing (ANLP)*, 194–201. 4.2.2

ROBERTSON, S. (1977). The probability ranking principle in information retrieval. *Journal of Documentation*, **33**, 294–304. 2.4.2.3

ROBERTSON, S. (2004). Understanding inverse document frequency: On theoretical arguments for IDF. *Journal of Documentation*, **60**, 503–520. 2.3.2.2

Robertson, S. & Sparck Jones, K. (1976). Relevance weighting of search terms. *Journal of the American Society of Information Science*, **27**, 129–146. 2.4.2.3

Robertson, S. & Walker, S. (1994). Some simple approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proceedings of the Seventeenth Annual International Association for Computing Machinery (ACM) SIGIR Conference on Research and Development in Information Retrieval*, 232–241, Springer-Verlag. 2.4.2.3, 2.4.2.3.1, 2.4.2.3.1, 2.4.2.3.1, 7.2.2.3, 7.2.2.4, 7.2.3.3, 7.3.3.4.3, 7.3.3.4.4, 7.3.3.4.4

Robertson, S.E. (1990). On term selection for query expansion. *Journal of Documentation*, **46**, 359–364. 2.5.1

Rocchio, J.J. (1971). Relevance feedback in information retrieval. In *The SMART Retrieval System– Experiments in Automatic Document Processing*, Prentice Hall. 2.5.1

Rosenfeld, R. (1994). *Adaptive Statistical Language Modeling: a Maximum Entropy Approach*. PhD thesis, Carnegie Mellon University. 3.4.2

Ross, J.R. (1973). Nouniness. *Three Dimensions of Linguistic Research*, 137–257. 3.3

Salton, G. (1962). The use of citations as an aid to automatic content analysis. Tech. rep., Harvard University. 2.4.2.2

Salton, G. (1966). Automatic phrase matching. *Readings in Automatic Language Processing*, 169–188. 2.5.2

Salton, G. (1971). *The SMART Retrieval System - Experiments in Automatic Document Processing*. Prentice-Hall. 2.1

Salton, G. (1972). A new comparison between conventional indexing (MEDLARS) and automatic text processing (SMART). *Journal of the American Society for Information Science*, **23**. 1

Salton, G. (1988). Syntactic approaches to automatic book indexing. In *Meeting of the Association for Computational Linguistics*, 204–210. 3.5

Salton, G. & Buckley, C. (1980). Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, **41**, 28–297. 2.5.1

Salton, G. & Buckley, C. (1988). Term weighting approaches in automatic text retrieval. *Information Processing and Management*, **24**, 513–523. 2.4.2

Salton, G. & McGill, M. (1983). *Introduction to Modern Information Retrieval*. McGraw-Hill. 2.1, 2.4.2.1

Salton, G., Wong, A. & Yang, C.S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, **18**, 613–620. 2.4.2.2

Salton, G., Buckley, C. & Yu, C.T. (1983). An evaluation of term dependence models in information retrieval. In J.J. Kuehn, ed., *SIGIR*, 151–173. 2.5.2

Santini, M. (2007). *Automatic Identification of Genre in Web Pages*. Phd thesis, University of Bristol. 4.2.2, 4.3.3

Santorini, B. (1990). Part-of-speech tagging guidelines for the Penn Treebank Project. Tech. rep., University of Pennsylvania. 3.4

Savicky, P. & Hlavacova, J. (2002). Measures of word commonness. *Journal of Quantitative Linguistics*, **9**, 215–231. 3.5, 8.2.5

Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing*, 44–49. 3.4.3, 6.3.1

Schmid, H. (1997). Probabilistic part-of-speech tagging using decision trees. *New Methods in Language Processing Studies*, 154–164. 4.2.2

Schuegraf, E.J. & Heaps, H.S. (1973). Selection of equifrequent word fragments for information retrieval. *Information Storage and Retrieval*, **9**, 697–711. 4.2.2

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, **34**, 1–47. 4.3.3

Shannon, C. (1948). A mathematical theory of communication, Part I. *Bell System Technical Journal*, **27**, 379–423. 4.2.2

Shneiderman, B. (1997). *Designing the User Interface: Strategies for Effective Human-computer Interaction*. Addison-Wesley. 2.2

SIGIR 1987 (1987). *Proceedings of the 10th Annual International Association for Computing Machinery (ACM) SIGIR Conference on Research and Development in Information Retrieval, New Orleans, USA, 1987*. E

Sigurd, B., Eeg-Olofsson, M. & van Weijer, J. (2004). Word length, sentence length and frequency - Zipf revisited. *Studia Linguistica*, **58**, 37–52. 6.3.2.2

Sinclair, J. (1991). *Corpus Concordance Collocation*. Oxford University Press. 1

Singhal, A., Buckley, C. & Mitra, M. (1996). Pivoted document length normalization. In Frei *et al.* (1996), 21–29. 2.4.2.2

Smeaton, A.F. (1986). Incorporating syntactic information into a document retrieval strategy: An investigation. In *SIGIR*, 103–113. 2.5.2, 3.5

Smeaton, A.F. (1999). *Using NLP or NLP Resources for Information Retrieval Tasks. Natural Language Information Retrieval*. Kluwer Academic Publishers. 3.5

Smeaton, A.F. & van Rijsbergen, C.J.K. (1988). Experiment on incorporation syntactic processing of user queries into a document retrieval strategy. In Chiaramella (1988), 31–51. 3.5

Smith, F.J. & Devine, K. (1985). Storing and retrieving word phrases. *Information Processing Management*, **21**, 215–224. 2.5.2

Soffer, A. (1997). Image categorization using texture features. In *4th International Conference Document Analysis and Recognition (ICDAR)*, 233–237, IEEE Computer Society. 4.2.2

Song, F. & Croft, W.B. (1999). A general language model for information retrieval. In Gey *et al.* (1999), 279–280. 2.4.2.4, 2.5.2

Sparck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, **28**, 11–21. 1.2, 2.3.2.2

Sparck Jones, K. (1999). Information retrieval and artificial intelligence. *Artificial Intelligence*, **114**, 257–281. 3.5

Sparck Jones, K. & Tait, J. (1984). Linguistically motivated descriptive term selection. In *22nd Annual Conference on Computational Linguistics (COLING)*, 287–290. 3.5

Sparck Jones, K. & van Rijsbergen, C.J.K. (1976). Information retrieval test collections. *Journal of Documentation*, **32**, 59–75. 2.2

Sparck Jones, K. & Willett, P. (1997). *Readings in Information Retrieval*. Morgan Kaufmann. 2.1

Sparck Jones, K., Walker, S. & Robertson, S.E. (2000). A probabilistic model for information retrieval: development and comparative experiments. (parts 1 & 2). *Information Processing and Management*, **36**, 779–840. 2.4.2.3

Srikanth, M. & Srihari, R.K. (2003). Incorporating query term dependencies in language models for document retrieval. In *SIGIR 2003: Proceedings of the 26th Annual International Association for Computing Machinery (ACM) SIGIR Conference on Research and Development in Information Retrieval, July 28 - August 1, 2003, Toronto, Canada*, 405–406. 3.5

Stiles, H.E. (1961). The association factor in information retrieval. *Journal of the ACM*, **8**, 271–279. 2.5.2

Strzalkowski, T. & Lin, F. (1997). Natural language information retrieval TREC-6 report. In Voorhees & Harman (1997), 347–366. 2.3.1.1, 3.5

Strzalkowski, T. & Sparck Jones, K. (1996). NLP track at TREC-5. In Voorhees & Harman (1996), 97–102. 3.5

Suen, C.Y. (1979). N-gram statistics for natural language understanding and text processing. *Institute of Electrical and Electronics Engineers (IEEE) Transactions on Pattern Analysis and Machine Intelligence*, **1**, 164–172. 4.2.2

Tait, J.I., ed. (2005). *Charting a New Course: Natural Language Processing and Information Retrieval. Essays in Honour of Karen Sparck Jones*. Springer. 3.5

Teng, C.Y. & Neuhoff, D.L. (1995). An improved hierarchical lossless text compression algorithm. In *Data Compression Conference*, 292–301. 4.2.2

Tuldava, J. (1996). The frequency spectrum of text and vocabulary. *Journal of Quantitative Linguistics*, **3**, 38–50. 5.2

Turtle, H.R. & Croft, W.B. (1991). Evaluation of an inference network-based retrieval model. *ACM Transactions on Information Systems*, **9**, 187–222. 2.4.2, 2.5.2

Ullmann, J.R. (1977). Binary n-gram technique for automatic correction of substitution, deletion, insertion, and reversal errors in words. *The Computer Journal*, **20**, 141–147. 4.2.2

van Rijsbergen, C.J.K. (1977). A theoretical basis for the use of co-occurrence data in information retrieval. *Journal of Documentation*, **33**, 106–119. 2.5.2

van Rijsbergen, C.J.K. (1979). *Information Retrieval*. Butterworths. 2.1, 2.4.2.2, 2.7

van Rijsbergen, C.J.K. (1986). A non-classical logic for information retrieval. *The Computer Journal*, **29**, 481–485. 2.4.2

van Rijsbergen, C.J.K., Harper, D. & Porter, M. (1981). The selection of good search terms. *Information Processing and Management*, **17**, 77–91. 2.5.1

Voorhees, E. & Harman, D. (1998). Overview of the 7th text retrieval conference. In E.M. Voorhees & D.K. Harman, eds., *TREC*, 1–24, NIST. 3.5

Voorhees, E.M. & Harman, D.K., eds. (1996). *NIST Special Publication 500-238: The Fifth Text REtrieval Conference (TREC-5)*, NIST. E

Voorhees, E.M. & Harman, D.K., eds. (1997). *NIST Special Publication 500-240: The Sixth Text REtrieval Conference (TREC-6)*, NIST. E

Voorhees, E.M. & Harman, D.K., eds. (2001). *NIST Special Publication 500-250: The Tenth Text REtrieval Conference (TREC 2001)*, NIST. 7.3.2.3.1, 7.3.2.3.4, E

White, O., Dunning, T., Sutton, G., Adams, M., Venter, J.C. & Fields, C. (1993). A quality control algorithm for DNA sequencing projects. *Nucleic Acids Research*, **21**, 3829–3838. 4.2.2

WILLETT, P. (1979). Document retrieval experiments using indexing vocabularies of varying size. II. Hashing, truncation, digram and trigram encoding of index terms. *Journal of Documentation*, **35**, 296–305. 4.2.2

WISKIEWSKI, J.L. (1987). Effective text compression with simultaneous digram and trigram encoding. *Journal of Information Science: Principles and Practice*, **13**, 159–164. 4.2.2

WITTEN, I.H., MOFFAT, A. & BELL, T.C. (1999). *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kaufmann Publishing. 2.6, 2.6.1

XU, J. & CROFT, W.B. (1996). Query expansion using local and global document analysis. In Frei *et al.* (1996), 4–11. 2.3.1.2, 2.5.1

YU, C.T. & SALTON, G. (1976). Precision weighting - an effective automatic indexing method. *Journal of the ACM*, **23**, 76–88. 2.3.2.2

YU, C.T., BUCKLEY, C., LAM, K. & SALTON, G. (1983). A generalised term dependence model in information retrieval. *Information Technology: Research and Development*, **2**, 129–154. 2.5.2

ZAMORA, E.M., POLLOCK, J.J. & ZAMORA, A. (1981). The use of trigram analysis for spelling error detection. *Information Processing and Management*, **17**, 305–316. 4.2.2

ZHAI, C. (1997). Fast statistical parsing of noun phrases for document indexing. In *Applied Natural Language Processing (ANLP)*, 312–319. 3.5

ZHAI, C. & LAFFERTY, J.D. (2001). A study of smoothing methods for language models applied to ad hoc information retrieval. In Croft *et al.* (2001), 334–342. 2.4.2.4, 2.4.2.4, 2.4.2.4

ZHAI, C., TONG, X., MILIC-FRAYLING, N. & EVANS, D. (1997). Evaluation of syntactic phrase indexing - CLARIT NLP track report. In Voorhees & Harman (1996), 335–340. 3.5

ZHANG, J. & MADDEN, T.L. (1997). PowerBLAST: A new network BLAST application for interactive or automated sequence analysis and annotation. *Genome Research*, **7**, 649–656. 4.2.2

Zhang, M., Lin, C. & Ma, S. (2004). How effective is query expansion for finding novel information? In K.Y. Su, J. ichi Tsujii, J.H. Lee & O.Y. Kwong, eds., *IJCNLP*, 149–157. 2.5.1

Zipf, G.K. (1949). *Human Behaviour and the Principle of Least Effort*. Addison-Wesley Press. 2.3.2.2, 6.3.2.2

Zobel, J. & Moffat, A. (2006). Inverted files for text search engines. *Computer Surveys*, **38**, 215–231. 2.3, 2.3.1, 2.3.1.1, 2.3.2.1, 2.3.2.2, 2.4.1, 2.4.2, 2.4.2.1, 2.4.2.2, 2.4.2.4, 2.5.2, 2.6, 2.6.1, 2.6.2, 2.7

Zobel, J., Moffat, A. & Ramamohanarao, K. (1996). Guidelines for presentation and comparison of indexing techniques. *Special Interest Group on Management of Data (SIGMOD) Record*, **25**, 10–15. 2.3.2.1

Zobel, J., Moffat, A. & Ramamohanarao, K. (1998). Inverted files versus signature files for text indexing. *Transactions Database Systems*, **23**, 453–490. 2.4.2

Zubov, A. (2004). Formalization of the procedure of singling out of the basic text contents. *Journal of Quantitative Linguistics*, **11**, 33–48. 3.5, 8.2.5