Automatic detection of new domain-specific words, using document classification and frequency profiling

Jørg Asmussen
Department for Digital Dictionaries and Text Corpora
Society for Danish Language and Literature, DSL
ja@dsl.dk

Corpus Linguistics 2005, draft version

Abstract

This paper describes and discusses an approach for automatically determining new domain-specific words, implemented at the Society for Danish Language and Literature, DSL, to facilitate updating The Danish Dictionary, DDO. The approach uses 'known' domain-specific vocabularies derived from the Corpus of the Danish Dictionary, DDOC, which are being used for domain-classifying new 'unknown' text material by comparing the vocabulary of a text to be classified with each domain-specific vocabulary and selecting the domain with the best match. New domain-specific vocabulary is then detected by statistically determining salient words in the new text material which are not yet registered in the dictionary.

1 Background: updating Den Danske Ordbog

The compilation of corpora and corpus-based dictionaries are major activities at the Society for Danish Language and Literature, DSL.

After having completed *Den Danske Ordbog*, 'The Danish Dictionary', [DDO: Hjorth et al., 2003] a new corpus-based written-from-scratch printed dictionary of contemporary Danish, cf. [Lorentzen, 2004], DSL is currently preparing a web-based version of it to which the newest vocabulary is continuously added. In order to rationalise the task of writing new dictionary entries, new vocabulary is subdivided into domains and treated by lexicographers specialised within one or more of these domains.

The source of new vocabulary is text material of various kinds, mainly taken from newspapers and other periodicals, material which is automatically transferred to DSL from a large Danish media database, *www.infomedia.dk*. From this daily incoming material possible new words are extracted as candidates for inclusion in the dictionary.

Consequently, the task is to assign each of the incoming texts to a particular domain and to extract from its vocabulary words so far not listed in the dictionary and treat them as candidates for inclusion as new domain-specific words. In order to accomplish this task, two methodological issues need to be addressed:

1. The design of a suitable domain classification in terms of granularity (how many different domains?), and contents (how can we define a certain domain intensionally, e.g. by its typical vocabulary?)

2. The classification procedure: on what methodological grounds can we assign a certain text to a particular domain?

There is certainly no decisive solution to the first issue. The most adequate solution would probably be to analyse a large number of texts statistically in order to get text clusters sharing some significant vocabulary and subsequently put domain labels on these clusters. The number of domains would then be the number of clusters, and the typical vocabulary of that domain could, for example, be determined by comparing the texts belonging to that domain with the corpus as a whole.

Another solution is just to choose a convenient number of domains as was done in [Jørgensen et al., 2003] where an approach to building Danish language corpora for certain domains was described. The number of domains is rather small and appears quite arbitrary: four domains (Information Technology, Environment, Health, and Public Administration) were defined, a total of six were planned.

The approach taken at DSL is to stick to a pre-established domain categorisation: the decimal classification system *DK5* which is basically derived from Dewey's Decimal Classification and which is used by public libraries in Denmark, cf. [Friis-Hansen, 1978]. As this classification is organised hierarchally, it allows various degrees of granularity. Hence, in our approach, the number of domains is set significantly higher than six in order to be better able to grasp some aspects of domain-dependent lexical semantics as well, cf. section 5.

Concerning the classification procedure proper, [Jørgensen et al., 2003] employ a catalogue of "relevant search words" which were used in web-searches in order to match appropriate text material. The search words (termed the "onomasiological structure") were manually selected from domain-specific thesauri and literature. A text containing an (unspecified) number of such words belonging to

the "onomasiological structure" of a certain domain was assigned to that domain. Apart from this approach being time-consuming because one has to establish the "onomasiological structure" manually, the outcome of it may be problematic due to idiosyncracies: Which words are appropriate for manual selection from the thesauri, and which are not? How frequently must words from the "onomasiological structure" of a certian domain occur in a text before the text can be expected to belong to that domain?

The approach described in this paper also uses designated words which are assumed to be domain-specific to match against the vocabulary of unclassified texts, but these words are statistically drawn from a large corpus of Danish. We believe that this approach is less likely to be skewed by accidental decisions made by scholars having their own ideas about which particular words characterise a certain domain, even though it has its shortcomings as well, cf. section 6. The corpus which we will be using in this approach is the Corpus of the Danish Dictionary, DDOC, where every domain-specific text has a domain label attached to it, taken from a subset of DK5, comprising 66 domains.

2 Data: the Corpus of the Danish Dictionary

The Corpus of the Danish Dictionary, DDOC, comprises about 43000 text samples totalling around 40 million words. It was compiled by DSL 1991-93 and covers a broad variety of Danish language from the 'decade' 1983-1992. Each text sample in the corpus is preceded by a header giving detailed information concerning the text and its author(s). The header holds approximately 30 different types of information: apart from general bibliographic data, it lists, for example, whether the sample is taken from written or spoken language, LSP or LGP, the year of publication, the publishing medium, the genre, the domain or topic, or the sex and age of its author. For all non-fiction text samples (88.6%) in the corpus, the general domain or topic is indicated by one of 66 possible values, for example, Chemistry, Travelling, Psychology, Housing, Gardening, Health, etc. A detailed description of the DDOC and its design can be found in [Norling-Christensen and Asmussen, 1998].

From this material 66 different vocabularies were extracted which are assumed to be specific for each of the 66 domains in question. These domain-specific vocabularies are then used to classify unseen texts.

3 Analysis: deriving domain-specific vocabularies from the DDOC

The main strategy of the approach applied here is to keep it simple by avoiding any kind of textual preprocessing, for example, tagging or parsing of the corpus material. All that is needed is a rather primitive tokeniser which in our case splits on blanks, removes punctuation, transforms uppercase letters into lowercase, and transforms sequences of digits into one *X*.

The approach itself is based on comparing frequency profiles derived from domain-specific sub-corpora with the DDOC as a whole in order to detect significantly over-represented types within a certain domain.

The statistical significance test which is used for the comparison is log likelihood. The motivation to use log likelihood is that it yields acceptable results within a corpus comparison framework (cf. [Kilgarriff, 2001]) and is straightforward to implement (cf. [Garside and Rayson, 2000]).

The derivation method comprises the following three steps:

1. Creation of subcorpora: based on the domain codes given in the headers of the text samples in the corpus, a subcorpus for each of the 66 domains is created.

2. Creation of frequency profiles: for each of the 66 subcorpora a frequency profile is created, that is, the corpus is transformed into a type-frequency vector. The same applies to the whole DDOC.

3. Comparison of frequency profiles: by applying the log likelihood test, each of the 66 frequency profiles is compared with the frequency profile of the total DDOC. Types that are significantly ($p \geq 0.99$) over-represented within one of the domain-specific vocabularies compared to the vocabulary of the DDOC as a whole are listed and the lists are sorted descendently by the salience of the types within the domain, that is, by their log likelihood score. Each of these lists is assumed to contain the domain-specific vocabulary *D* of one of the 66 domains investigated.

A closer investigation of the content of each of these assumed domain-specific vocabularies reveals that they intuitively seem quite reasonable: figure 1 gives this impression by showing the top fifteen types for the domains Computing, Philosophy, and Economy, that is, the most salient types within these domains. Generally, the designation of the domain proper is among the top fifteen types.

However, even if the contents of the domain-specific vocabularies seem to be quite convincing at first glance, the approach is characterised by a couple of uncertainties.

$D_{computing}$	$D_{philosophy}$	$D_{economy}$
data	mennesket 'man'	kr Danish unit of currency (abbreviated)
programmer 'programs'	kierkegaard	X,X amount (with decimals)
computer	moral	pct
computeren 'the computer'	løgstrup Danish philosopher	procent 'percent'
edb 'computing'	aristoteles	kroner Danish unit of currency
computere 'computers'	filosofi	rente 'interest'
ibm	fornuft 'ratio'	offentlige 'public'
pc	platon	økonomiske 'economic'
kan 'can'	kierkegaards	bank
mb	tim probably a name	X figure, number
apple	den 'the'	økonomi 'economy'
amiga	menneskets 'man's'	vil 'will, shall'
commodore	filosof 'philosopher'	mia
windows	liv 'life'	milliarder
datamaskine 'computer'	sansning 'perception'	indkomst earnings

Figure 1: Most salient types for three domain-specific vocabularies

The first uncertainty concerns the chosen significance level, here $p \geq 0.99$. A different threshold would have given more or fewer types within each domain. The arbitrary choice of such a level of significance in connection with differently sized domain-specific subcorpora also causes the different domain-specific vocabularies to be of different size. Thus, the domain Folklore has a specific vocabulary of only 1957 types whereas the Sport domain comprises 16022 types, the average for all 66 domains being 7256. Another approach could have focused on equally sized domain-specific corpora or equally sized domain-specific vocabularies, for example, the hundred most salient types for each domain, no matter how big the underlying corpus was. We consider statistical significance as a more reliable measure than an equal – but arbitrarily chosen – number of types for each domain-specific vocabulary. However, classifying unknown texts on the basis of our assumed domain-specific vocabularies has to take into account that the size of these vocabularies varies and thus may bias the classification of texts on these vocabularies.

The second uncertainty is frequent function words appearing so saliently ranked in our domain-specific vocabularies. As can be seen in figure 1, *den* ('the') and *vil* ('will') appear rather highly ranked in the tables. The same applies to a considerable amount of other function words appearing in other domains or lower ranked in the three domains shown. This phenomenon clearly runs counter to

intuition of the significance of these words for a certain domain. However, these types are not excluded from the vocabularies as they may in some cases be part of fixed expressions which may be significant for a certain domain.

4 Classification: matching unknown text against known domain-specific vocabularies

Now that 66 assumed domain-specific vocabularies have been determined, the next step is to assign unseen texts to one of these domains. The starting point for our classification approach is the concept of the largest intersection $|D \cap T|$ between the set of types of each of the 66 domain-specific vocabularies and the set of types T of the text to be classified. The more types the text has in common with the domain-specific vocabulary of a certain domain, the more likely it is that the text belongs to that domain. The simplest way to implement this is to extract a table of all types from the text, take each type from this table and compare it with each type from each of the domain-specific vocabularies: the one with most types in common with the text to be classified wins, that is, the text will be assigned to the according domain.

However, this concept needs to take into account a couple of textual properties. If we look at a certain text of, say, 50 words that has one occurrence of each of the 15 types in the Computing column in figure 1, $|D \cap T| = 15$ which means that it probably would be classified as a Computing text. If another same-sized text has, e.g., 10 occurences of *computer*, 5 of *data*, and 4 of *programmer*, $|D \cap T| = 3$ which would make it more unlikely that this text also would be classified as belonging to the Computing domain even if it actually were as 'computingish' as the first one. Hence, the frequency of types in a text should be taken into consideration in the classification process as well. This can be achieved by considering the intersection between D and the set of *tokens*, W, in the text to be classified $|D \cap W|$ instead of $|D \cap T|$.

As already mentioned in the previous section, the 66 domains are not evenly distributed in the DDOC, but have significantly varying sizes which are consequently reflected by the sizes of the domain-specific vocabularies. The compositional strategy during the compilation of the DDOC was to have as many different domains as possible represented in the corpus, as a weak representation was assumed better than none at all. Hence, the resulting differently sized domain-specific vocabularies have implications on the text classification approach: it is more likely that a domain with a large vocabulary wins over weaker domains, that is, its $|D \cap W|$ is greater, thus making it more likely that a text will be classified as a Sport text than as a Folklore text. Therefore, domains with large vocabular-

ies would be privileged if we used the largest intersection approach without any further modifications. A possible solution to compensate for the lack of balance, is not just to count the number of elements, that is, tokens, in the intersection, but to weigh these counts relatively with the size of the domain-specific vocabulary for comparison. Thus, one type from the Sport domain could count as $\frac{1}{16022}$, whereas each Folklore type could count relatively more, namely $\frac{1}{1957}$. However, experiments show that the inverse of $|D|$ gives too much weight to small domains whereas $\frac{1}{\sqrt{|D|}}$ performs better.

Bias may also arise from the large amount of function words in the domain-specific vocabularies as already mentioned in section 3. Function words and other intuitively less typical domain-specific words often occur in many different domains. To compensate for their getting too much weight, the number of domains a given type belongs to could be taken into consideration as well when a domain score is being computed. A weight which can compensate for words belonging to too many domains is the inverse of the number of domains a certain word belongs to: $\frac{1}{d}$.

Another aspect which may be important for computing a domain score is the rank of a certain word within a domain-specific vocabulary. Thus, a higher ranked word could count more than a lower ranked one. However, the relevance of this factor has not yet been investigated.

Figure 2 lists all the parameters that seem relevant to compute a score for a particular domain.

	Parameter	Expression
1	A particular text token has to be member of the intersection between the set of all text tokens and the domain-specific vocabulary	$t \in D \cap W$
2	Weight based on the size of the domain-specific vocabulary of the domain in question	$v = \frac{1}{\sqrt{\lvert D \rvert}}$
3	Weight based on the number of domains a certain text token is a member of, its 'uniqueness'	$w = \frac{1}{d}$ for $d > 0$ $w = 0$ otherwise
4	Number of unknown tokens, i.e., tokens not matched with any type from any of the domain-specific vocabularies in the text to be classified	u same as $n - k$
5	Number of known tokens, i.e., tokens matched with at least one type from one of the domain-specific vocabularies	k same as $n - u$
6	The number of tokens in the unclassified text – only used to make the score relative to the text length	n same as $u + k$

Figure 2: Parameters to take into account for text classification

A general methodological issue is an adequate combination of these paramet-

ers into an expression that yields a viable score for each domain. Intuition-based experiments have shown that the following expression yields quite acceptable results:

$$s_D = \frac{1}{n} \cdot \frac{k}{u} \cdot v \cdot \sum_{t \in D \cap W} w_t$$

For each of the 66 possible domains D, the expression computes a score S_D based on the tokens of the text to be classified. Subsequently, the text is assigned to the domain with the highest score.

Figure 3 shows a small text sample to illustrate the approach. The text is an excerpt from a Danish website that describes how to install Linux on a PC, viz, an installation guide presumably belonging to the Computing domain. In the sample shown, every text token which is a member of the vocabulary specific for the Computing domain ($t \in D \cap W$, cf. row 1 in figure 2) has been indexed with the number of domain-specific vocabularies it is a member of (the d-value from row 3 in figure 2); tokens which do not belong to Computing-specific vocabulary ($t \notin D \cap W$) have been indexed with a dash.

Danish	English translation
Du<-> skal<27> bruge<19> en<32> diskette<1> til<31> installationen<5>. På<24> et<34> tidspunkt<9> bliver<31> du<-> spurgt<-> om<-> du<-> vil<26> lave<20> en<32> bootdiskette<->. Erfaringen<-> siger<-> at<-> det<-> godt<-> kan<36> betale<-> sig<-> at<-> formatere<-> en<32> diskette<1> i<-> forvejen<-> med<24> tjek<-> for<-> dårlige<-> sektorer<->. Før<-> du<-> installerer<1> Linux<->, skal<27> der<22> være<-> en<32> partition<-> til<31> rådighed<13>, der<22> er<40> stor<22> nok<8> til<31> at<27> rumme<10> det<-> hele<-> (samt<-> en<32> swap-partition<->). I<-> løbet<-> af<45> Linux-installationen<-> vil<26> der<22> blive<18> lejlighed<-> til<31> at<-> repartitionere<-> så<-> meget<->, du<-> har<24> behov<22> for<->, inden<21> for<-> den<-> plads<->, der<22> nu<-> er<40> blevet<-> til<31> rådighed<13>.	You will need a diskette for the installation. At a point you will be asked if you want to create a boot diskette. Experience shows that it is worthwhile to format a diskette in advance with a check for bad sectors. Before you install Linux, there must be allocated a partition which is big enough to contain everything (as well as a swap partition). During the Linux installation there will be an opportunity to repartition as much as you need within the space which now has been made available.

Figure 3: Sample text

The computation of a score for this text for the Computing domain comprises

the following steps:

1. All d-based weights for text tokens belonging to the Computing-specific vocabulary are summed:

$$\sum_{t \in D \cap W} w_t = \frac{1}{27} + \frac{1}{19} + \frac{1}{32} + \frac{1}{1} + \ldots + \frac{1}{22} + \frac{1}{40} + \frac{1}{31} + \frac{1}{12} \approx 4.5742$$

2. A weight based on the size of the domain-specific vocabulary, in this case Computing comprising 6277 types, is computed:

$$v = \frac{1}{\sqrt{|D|}} = \frac{1}{\sqrt{6277}} \approx 0.0126$$

3. The ratio between known (i.e., member of a domain-specific vocabulary) and unknown (i.e., not member of any domain-specific vocabulary) text tokens is determined:

$$\frac{k}{u} = \frac{74}{7} \approx 10.5714$$

4. A factor is determined relating the score to the size of the text, that is, the number of text tokens:

$$\frac{1}{n} = \frac{1}{81} \approx 0.0123$$

5. Finally, the total score for the Computing domain is determined:

$$S_D = 0.0123 \cdot 10.5741 \cdot 0.0126 \cdot 4.5742 \approx 0.0075$$

Compared to the scores obtained for all other domains, the Computing score is the highest one. Consequently, the sample text is assigned to the Computing domain.

5 Comparison: determining new domain-specific vocabulary

In order to determine new domain-specific vocabulary for lexicographic description, a frequency profile of the automatically classified new corpus material is compared with a frequency profile of the DDOC. The comparison is once again based on the log likelihood test in order to determine the salient vocabulary of the new domain-specific corpus material compared with the DDOC as a whole. The size of a new portion of domain-specific corpus material should be approximately equal to the size of the domain-specific subcorpus being part of the DDOC. For the

Computing domain, this size is about 138000 tokens. As soon as newly classified material within this domain reaches this size, this material is compared with the DDOC, and salient new words within this domain are determined and proposed for inclusion in the DDO if not already included. It does not make much sense to process a single text in this way because due to its restricted length it is likely to bias the result, overemphasising words which may be salient for this text only, but which do not have any lexicographic relevance for the domain. However, to demonstrate the mechanism behind this approach and to discuss some of its results, we will compare our 81 token text sample with the DDOC. Figure 4 shows the most salient types in this sample ($p \geq 0.999$), ranked by saliency, together with their frequency in the DDOC (f_{DDOC}), their frequency in the sample (f_{sample}), and an indication of which domains their sense definitions given in the DDO belong to. Every sense definition in the DDO carries a domain indication; however, this indication is for internal use only, that is to say not visible in the printed version of the dictionary. The domain classification system applied here is identical to that used in the DDOC. The types listed in the table need to be lemmatised in order to be automatically compared with the corresponding headwords in the DDO – in the current example we just lemmatise and look up the corresponding entries in the dictionary manually.

As can be seen, the most salient word in the sample text compared to the DDOC as a whole is *diskette*. This word is already defined in the DDO as a Computing word. The same applies to *formatere*. Some words (*bootdiskette*, *Linux*, *Linux-installation*, *partition*, *repartitionere*, *swap-partition*) do not have any entry in the DDO at all. In these cases the lexicographer responsible for this domain would have to decide which words need to be included in the dictionary. Words which definitely not should be included, for example, proper nouns or misspellings, can be blacklisted by the editor, so they are permanently deleted from all future lists of potential candidates. A new word likely to be included from the list could be *partition*. The editor now knows that this word is salient in the Computing domain and probably has a Computing sense.

More interesting candidates on the list are *installere*, *installation*, and *sektor*. All these words already have entries in the DDO, but none of them seems to have a special Computing definition. In these cases the editor should check if one of the given definitions already covers a potential Computing sense, otherwise he should decide whether to include an extra Computing definition – the latter should actually be the case for the three mentioned candidates as none of the already existing definitions actually covers the Computing sense.

Candidates such as *rådighed*, *du*, *tjek*, *erfaring*, and *rumme* are a little more tricky as they introspectively do not seem typical for the Computing domain. Words like these would not occur so prominently ranked if we had compared a larger collection of Computing texts with the existing DDOC. Hence, they would

Type	f_{DDOC}	f_{sample}	DDO domains
diskette	78	2	*Computing*
bootdiskette	0	1	missing entry
formatere 'format'	0	1	*Computing*
linux	0	1	missing entry
linux-installationen 'the linux installation'	0	1	missing entry
partition	0	1	missing entry
repartitionere 'repartition'	0	1	missing entry
swap-partition	0	1	missing entry
rådighed 'disposition'	1730	2	*General*
installerer 'install(s)'	16	1	*General* *Technology*
du 'you'	143798	5	*General*
installationen 'the installation'	34	1	*Technology* *Art* *Military*
tjek 'check'	100	1	*General*
sektorer 'sectors'	112	1	*Society* *Politics* *Mathematics*
erfaringen 'the experience'	217	1	*General* *Psychology* *Philosophy*
rumme 'contain'	480	1	*General* *Physics* *Nautics*

Figure 4: Most salient types in the sample text

not appear in the real-life implementation of this approach.

6 Discussion and future work

This paper presented an approach for automatically determining new domain-specific words for lexicographic description. The approach comprises the fol-

lowing steps:

1. On the basis of an existing corpus, domain-specific vocabularies are derived.
2. The domain-specific vocabularies are used for classifying new text material.
3. Domain-classified new material is compared with the original corpus and salient words in the new material are processed as candidates for new entries/definitions.

Each of these steps involves some basic decisions which unintentionally may influence the outcome.

Concerning the first step, we adopted the domain classification system used in the DDOC as it was. Alternatively, we could have analysed its appropriateness for our particular task. The relatively high number of domains and the greatly varying amount of text found under each domain may reduce the quality of the text classification system based on it. Alternatively, we could have redesigned the classification system by for example merging domains with only few texts in order to get the variance of the quantity of texts within each domain reduced and thus also the granularity of the system. This may improve the precision of the text classification.

The domain-specific vocabularies were derived from the DDOC by applying the log likelihood test. Other tests may perform better, for example the Mann-Whitney ranks test, cf. [Kilgarriff, 2001]. Here, we just chose a compromise between the most straightforward implementation and a test that yields acceptable results in a linguistic context. But our choice may be arbitrary from a strictly linguistic point of view as it does not necessarily reflect the nature of the object investigated.

Concerning the second step, the text classification, a score expression was designed that takes into account some properties of the material to be processed: the overlap of tokens in the unseen text and a certain domain-specific vocabulary, the size of the domain-specific vocabulary, the uniqueness of a certain type for a particular domain, and the ratio between recognised and unrecognised tokens in the unseen text. Other properties may be worth taking into account as well, for example, the salience rank of a word in a certain domain-specific vocabulary. Thus, our approach reflects a couple of assumptions concerning the quality of the object we are investigating, and it attempts to quantify them in order to get texts classified correctly – in regard to our own intuition. The results of the approach seem acceptable, but it is doubtful if the approach proper is acceptable too from a linguistic point of view. And it would be worthwile investigating if it really quantifies the nature of language adequately. Other approaches could have been chosen

such as decision trees or k Nearest Neighbour classification for the text classification task, cf. [Manning and Schütze, 1999]. But these approaches also involve arbitrary, intuition-based decisions that do not necessarily reflect regularities of language appropriately.

The third step in our approach, the extraction of new domain-specific vocabulary, is based on a log likelihood comparison of a newly gathered domain-specific corpus with the original corpus. Alternatively, we could have compared it with its domain-specific counterpart in the DDOC, but this type of comparison is likely to overemphasize general words in the new material. Again an instrospection-driven observation lacking any kind of proof.

The mutual dependencies between these decisions are complex, thus making it rather difficult to estimate the severity of each of them. However, future testing of various alternating parameters in the processing chain, could probably fine-tune and improve the approach. The classification part could be tested in a setup, where the DDOC is divided into two equally sized parts, each one comprising the same amount of text material from each domain. The one half would be used for deriving the domain-specific vocabularies and the second half for testing various methodological alternatives.

Another point which has to be addressed in future work is the fact that domain-specific vocabularies develop and change over time. Newly determined domain-specific vocabulary thus has to be taken into account when subsequently processing more text material. Two approaches seem obvious, namely either to add the newly determined domain-specific to the already existing vocabulary, or to add the newly gathered domain-specific corpus to the DDOC and to re-run the determination of domain-specific vocabularies which will subsequently be used for further text classification. Testing must show which of these approaches performs best.

However, to summarise it can be ascertained that the mechanism behind our approach is applicable for the described task, but it does not quantitatively *explain* what makes a word or a text domain-specific – nor does it explain what makes a word new. It works acceptably, is implementable, but it does not shed much light on the nature and regularities of language; and it does not become less introspective just because it is quantitative. A property that it seems to share with many other quantitative approaches within linguistics.

References

[DDO: Hjorth et al., 2003] DDO: Hjorth, E., Kristensen, K., Lorentzen, H., Trap-Jensen, L., Asmussen, J., et al., editors (2003). *Den Danske Ordbog 1-6*. DSL & Gyldendal, København/Copenhagen.

[Friis-Hansen, 1978] Friis-Hansen, J. B. (1978). *Hjælpebog til DK5*.

[Garside and Rayson, 2000] Garside, R. and Rayson, P. (2000). Comparing corpora using frequency profiling. In *Proceedings of the workshop on Comparing Corpora, held in conjunction with the 38th annual meeting of the Association for Computational Linguistics (ACL 2000)*, pages 1–6, Hong Kong.

[Jørgensen et al., 2003] Jørgensen, S. W., Hansen, C., Drost, J., Haltrup, D., Braasch, A., and Olsen, S. (2003). Domain specific corpus building and lemma selection in a computational lexicon. In Archer, D., Rayson, P., Wilson, A., and McEnery, T., editors, *Proceedings of the Corpus Linguistics 2003 conference*, pages 374–383, Lancaster. UCREL, Lancaster University.

[Kilgarriff, 2001] Kilgarriff, A. (2001). Comparing Corpora. *IJCL*, 6(1):97–133.

[Lorentzen, 2004] Lorentzen, H. (2004). The Danish Dictionary at large: presentation, problems and perspectives. In *Proceedings of the 11th EURALEX International Congress*, volume 1, pages 285–294, Lorient. Euralex.

[Manning and Schütze, 1999] Manning, C. D. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts, 2003 edition.

[Norling-Christensen and Asmussen, 1998] Norling-Christensen, O. and Asmussen, J. (1998). The Corpus of The Danish Dictionary. *Lexikos. Afrilex Series*, 8:223–242.