SCIENTIFIC PUBLICATIONS OF THE STATE UNIVERSITY OF NOVI PAZAR
SER. A: APPL. MATH. INFORM. AND MECH. vol. 9, 1 (2017), 1-18.

Topic Modelling with Morphologically Analyzed Vocabularies

Marcus Spies

Abstract: Probabilistic topic modeling is a text mining technique that allows to extract sets of term probability distributions which can intuitively be interpreted as latent topics. The extraction in most techniques uses only document term frequency matrices as input data. Moreover, topic models estimate posterior document-topic distributions useful for intelligent document retrieval query processing. This paper discusses two approaches to topic modeling involving Dirichlet distributions and Dirichlet processes.

However, these and related approaches presume suitable text preprocessing in order to keep parameter spaces for estimations from training text corpora at manageable sizes. In the present paper, we discuss the influence of morphological preprocessing of training texts. Morphological analysis is a computer linguistic discipline that allows to decompose observed terms into base lemmata. This is effected by a deep analysis of the observed terms as opposed to straightforward prefix or postfix elimination used in conventional stemming algorithms. Morphological preprocessing is especially effective in inflection rich languages like, e.g. Finnish or German, and effectively reduces the training vocabulary size. In addition, morphological preprocessing allows for decomposing compound words.

It is of considerable interest to study the influence of morphological preprocessing on text mining and statistical topic models. In experiments reported in the application section of this paper, significant changes of the frequency structure of document term matrices were found. Interestingly, these changes also led to substantial improvements in model quality indicators of topic models due to morphological preprocessing.

Steps for further research are suggested in the concluding section.

Keywords: computational morphologies, statistical topic models, latent semantic analysis, latent Dirichlet allocation, hierarchical Dirichlet processes, natural language processing

1 Probabilistic Topic Modeling

To start, we recall a common assumption underlying many information retrieval and almost all topic modelling techniques. It is the BOW (bag of words) assumption [19] claiming that

- A corpus is a set of documents.

Manuscript received September 15, 2016; accepted January 27, 2017.

Marcus Spies is Professor of Knowledge Management at LMU University of Munich, mspies@lrz.uni-muenchen.de

- A document is a bag of term occurrences (or tokens, often simply called words).

Corollaries of these claims are that relationships between documents are not relevant to information retrieval and that word sequences in documents can be adequately summarized by word counts. Obviously, these corollaries can be grossly inadequate, e.g. in poetic texts. However, for generic purposes of document characterization like in news services they suffice in most contexts.

Nex, key generic assumptions of probabilistic topic modelling are as follows –

- A topic is a discrete probability distribution over terms or items of a given vocabulary. This distribution is usually referred to as topic-term distribution. Topics are assumed to capture token co-occurrences in documents and form meaningful semantic aggregates of terms.

- A document topic model consists of an assignment of a topic to each term occurrence. Different occurrences of the same term may refer to different topics, allowing to capture polysemy of terms. Polysemy could be represented only through similarity relationships in earlier models based on latent semantic analysis [16].

This approach to topic modelling is attractive to content analytics since it does not presume apriori category knowledge and proposes to estimate topics by unsupervised learning. Applications to big data sets in organizational document retrieval are discussed in [25]. More specifically, in the currently most popular probabilistic modelling technique, the latent Dirichlet allocation (LDA, see [3]), the relevant assumptions from the above list are specialized as follows.

- LDA aims at approximating each document-term distribution by a document specific mixture distribution of a set k of components referred to as (latent) topics. k is the assumed number of latent topics and has to be found by experimentation or from prior knowledge. The document specific topic mixture distribution is commonly referred to as document-topic distribution.

- Each of the k mixture component distributions is a multinomial topic-term distribution (defined independently of documents).

- LDA relies on parametric Bayesian inference, generating the document-topic multinomials as posterior from a background prior Dirichlet distribiution. A common Dirichlet prior may be used for parametric Bayes inference on topic-term distributions, as well.

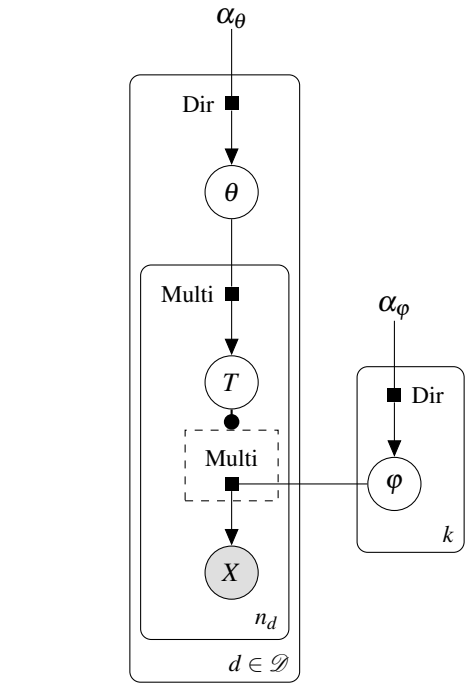
- Evaluation of results can be performed using posterior document likelihoods or perplexity values.

For an excellent introduction to Dirichlet distributions and their role in Bayesian parametric inference, see [9]. For LDA, variational inference [3] and Gibbs sampling [11] inference procedures have been implemented. LDA as a generative probabilistic approach

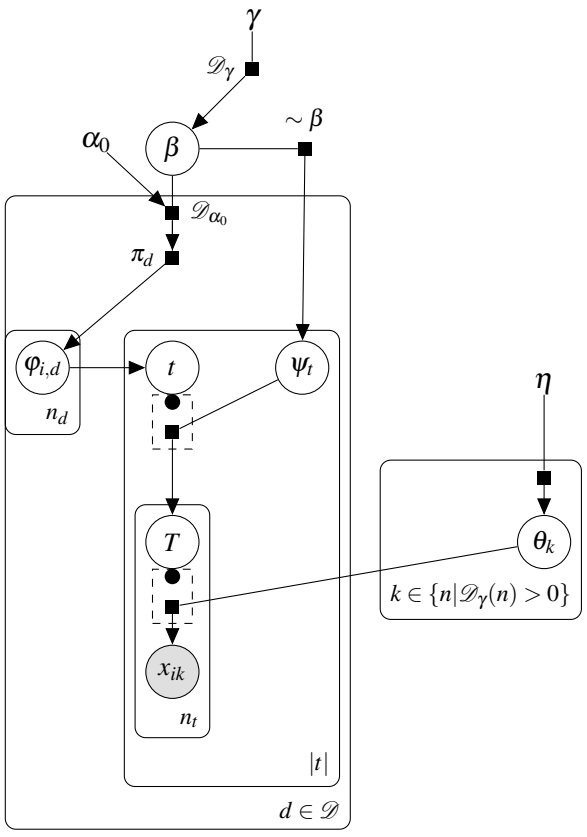
can be conveniently visualized using a combined plate / factor graph / gate model notation, in line with approaches from [13], [15] and [20], see figure 1(a), which is a slightly modified citation of a diagram provided by [6]. The interpretation of this graph is that the only observed variable X is the assignment of a token in $1,..,n_d$ from document d in $1,..,\mathscr{D}$ to a topic T in $1,...,k$ where the latent random variable T is multinomial and drawn from a document-topic distribution θ. θ, in turn, is drawn from a corpus wide Dirichlet distribution parameterized by a scalar α_θ, which controls its apriori dispersion. The latent topic T effectively gates draws from the selected topic-term distribution φ ranging over the vocabulary $\mathscr{T}$ and parameterized as multinomial using another Dirichlet prior.

In practice, an LDA solution is computed by combining MCMC techniques for sampling the pseudo-observations X with Bayesian parameter learning. Many libraries implementing LDA are available, some have been packaged in the `topicmodels` library for the `R` statistical programming language [12]. The `lda-c` package [1] (packaged into `topicmodels` for `R`) is based on the variational inference procedure as detailed in [3]. As example of a Gibbs sampler for LDA, we mention [26]. However, LDA has a number of properties which often turn into shortcomings in practical applications.

- Topics typically exhibit high commonality as overall frequent terms in a corpus also appear in many topics with high probabilities.
- Document topic proportions have often approximately singleton support, especially if documents are short. Multi-topic documents are not reliably analyzed.
- A common feature of all available implementations is that only symmetric Dirichlet priors, i.e. parameterized by a constant vector, are considered. With an appropriate choice of a scalar value for populating this vector, nearly arbitrary posterior distributions can be achieved. However, the limitation to a single posterior value might limit the scope of parameter inference.
- The choice of an appropriate topic number for a given corpus must be left with domain experts, as neither perplexity nor document likelihoods change monotonically with topic number.
- LDA solutions are highly non-unique. As shown by Vorontsov and Potapenko [29], LDA effectively computes a solution to a constrained but under-determined NNMF problem (non-negative matrix factorization). The authors propose an alternative regularized optimization approach imposing further desirable constraints on the possible solutions.



(a)



(b)

Fig. 1. An LDA factor and gate graph (after [6]), and a corresponding model graph for hierarchical Dirichlet process models as applied to text corpus analysis. For explanations and further references, see text.

2 Dirichlet process and Hierarchical Dirichlet process

In order to overcome the need to choose the number of topics in advance, a nonparametric Bayesian (i.e., parameterized by an unbounded set of parameters whose extent is determined dynamically) approach has been proposed in [28]. We briefly summarize key elements and provide a novel plate / gate diagram in fig. 1(b), which will be explained shortly. This diagram also allows a quick comparison to LDA.

We summarize key properties of a Dirichlet process [8],[24].

- A Dirichlet measure $\mathscr{D}_\alpha$ is a random probability measure on a measure space $(\mathscr{X}, \mathscr{B})$, parameterized by a finite measure $\alpha()$ on $(\mathscr{X}, \mathscr{B})$. $\alpha_0 = \alpha(\mathscr{X})$ is the measure allocated to the universe $\mathscr{X}$. As will become evident below, the higher α_0, the more dispersed probability measures drawn from $\mathscr{D}_\alpha$ will be.

- For every finite partition $B_1, \ldots, B_k$ of $\mathscr{X}$, the marginal distribution of $\mathscr{D}_\alpha$ is the finite Dirichlet distribution $\mathscr{D}_{\alpha(B_1),\ldots,\alpha(B_k)}$. $\mathscr{D}_\alpha$ supports discrete probability measures (pmf) with probability 1.

- The Bayesian posterior $\mathscr{D}_\alpha^X$ after observing a sample $X_i, i \in 1, \ldots, n$ is $\mathscr{D}_\alpha + \sum_{i=1}^n \delta_{X_i}$ (similar to a discrete Dirichlet posterior, see [8]).

- Based on the conditional distributions of sequential samples from $\mathscr{D}_\alpha$, sampling from $\mathscr{D}_\alpha$ can be simulated using the Chinese Restaurant process (CRP), see [9],[28, 2]. CRP is a clustering of integers modelled as guests arriving at a restaurant that offers countably many seats and tables. Seating the i-th guest at a table k is a metaphor for assigning i to cluster k. Let K_i denote the number of different tables after $i-1$ guests have been seated and m_k denote the occupancy numbers for those tables. Then, the assignment rule of CRP is to place arriving guest i at occupied table $X_k \in \mathscr{X}$ for $k \in \{1, \ldots, K_i\}$ with probability $\propto \sum_{k=1}^{K_i} m_k/(i-1+\alpha_0)$, at a new table else.

The CRP assignment rule shows that higher values of α_0 will decrease the probability of placing new guests at existing tables and thus increase the dispersion of the distribution of guests in a draw from the Dirichlet process. – Alternatively, based on the Beta-marginal of a Dirichlet distribution, sampling from $\mathscr{D}_\alpha$ can be simulated using a stick-breaking construction as introduced by Sethuraman [24].

Assuming $\mathscr{X}$ at least countably infinite, independent samples from a Dirichlet process share atoms with probability 0. In order to ensure sharing of elements between samples of a Dirichlet process, an extension to *hierarchical* Dirichlet processes has been proposed [28].

Basically, a Dirichlet process allows to estimate a Dirichlet mixture distribution model with a variable (principally unbounded) number of components, while a hierarchical Dirichlet process can be used to build such a model for grouped variables with shared components. In the terminology of probabilistic topic modelling, sharing of elements intuitively corresponds to sharing of topics across documents. This view opens the perspective for applying the approach to statisticall topic modelling with automated estimation of topic numbers.

The overall statistical model expressed by a hierarchical Dirichlet process as explained is summarized as a plate / gate diagram in fig. 1(b). Modelling conventions used in this diagram are the same as for fig. 1(a), and symbols are aligned with LDA nomenclature as far as reasonable. The explanation of the HDP model is summarized in the following items. We use double subscripts to make reading easier, while the diagram in fig. 1(b) uses single subscripts if the scope of the index is clear from the enclosing plate.

- HDP [28] aims at describing document term distributions by a mixture of an adaptively estimated number of component topic-term distributions indexed θ_k for a finite, however not apriori fixed or bounded integer k.
- A common corpus component or topic prior distribution is constructed from a base Dirichlet process $\mathscr{D}_\gamma$. Drawing from $\mathscr{D}_\gamma$ yields a sequence of proportions β_k modelling corpus wide topic probabilities.
- Document d topic proportions are derived from a secondary Dirichlet process $\mathscr{D}_{\alpha_0}$ instantiated for each d in $1,..,\mathscr{D}$. These secondary processes operate by random probability variations and component selections on $\mathscr{D}_\gamma$ yielding per document random measures π_d. For the derivation of the resulting distribution, see[28], p. 12 (eqs. 22-24).
- Tokens i within document d are clustered using the per document random measure π_d for drawing token-level random variables $\phi_{i,d}$. This is done for all n_d tokens in each document.
- It is helpful at this point to think of the $\phi_{i,d}$ as arranging tokens per document into virtual local topics corresponding to components drawn from the document Dirichlet process $\mathscr{D}_{\alpha_0}$. These virtual topics have been described in [28] and related publications as tables of a *Chinese restaurant franchise* combining a local Chinese restaurant process per document with a franchise provided selection of dishes. In the Chinese restaurant franchise metaphor, these virtual topics are referred to as tables in the local restaurant (document in our context) to which guests are assigned. The virtual topic or table to which a token is assigned is t.
- The corpus topic proportions β are used in a next step for drawing the topic selection for each virtual topic (or table) of tokens (guests) per document. This per-document per local cluster topic assignment is the value of the $\mathscr{D}_\gamma - \beta$-distributed random variables Ψ_t. In the Chinese restaurant franchise metaphor, this corresponds to the assignment of a global topic (franchise wide available dish) to a local or within-restaurant grouping of guests. By this assignment, the common components as pre-selected by the corpus wide Dirichlet process $\mathscr{D}_\gamma$ are now chosen as values of the local document components. The assignment as described is carried our for all local groups or tables of which we have $|t|$ in a given document.

- Note that β as drawn from the corpus wide $\mathscr{D}_\gamma$ process is a random probability measure. This is used as distribution of a factor generating the per-table topic selections Ψ_t. In the diagram in fig. 1(b), we use again the convention proposed by Laura Dietz [6] which allows a gate symbol with an iteration scope provided by the gating variable (in our case, t).

- Topic term distributions are constructed from a prior distribution parameterized η for each $k \in \{n|\mathscr{D}_\gamma(n) > 0\}$, i.e. for each k such that the corresponding numbered global component has non-zero probability The generation procedure for a given vocabulary of final and fixed size $|V|$ is usually a draw from a multinomial distribution for a sample size $|V|$ parameterized with a Dirichlet prior. As the number of terms is known and fixed for a text corpus analysis, no additional stochastic process is involved here.

- Gated by the topic drawn from Ψ_t, a draw from the corresponding topic-term distribution θ_k yields a term explaining the current observed token x_{ik}. This happens for all n_t tokens in the scope of a topic-to-table assignment t_i.

Regarding implementation, a Gibbs sampler based solution for HDP inference on a corpus of documents in the C++ language is available from [4], see [30] for the algorithmic foundations, which we do not dwell upon in this paper.

We now turn to applying hierarchical Dirichlet processes in topic modelling for natural language text corpora.

3 Morphological Analysis

Morphological analysis is a computer linguistic technique for decomposing words of a given natural language into basic components representing the word stem (or lemma) and any prefixes or suffixes. Morphological analysis also comprises segmentation of compound words and recursive (morphological) analysis of the composing words. Practical applications of morphologies in computer based text analysis usually also comprise part-of-speech (POS) annotation of the words appearing in a given text. Usually, especially in inflection rich languages like German, morphological analysis of a given word in context yields multiple results, including also multiple candidate POS tags. If an application requires choosing a single result, this will be typically computed using a sequential stochastic model like Hidden Markov (HMM) or Conditional Random fields (CRF), see [27]. In order to build a morphology of a given language, human effort by trained linguistic experts is needed, and additional statistical analyses of training data for estimating HMM or CRF parameters and optimizing the morphology are required.

Technically, most computer linguistic morphologies in use have been constructed using the formalism of finite state transducers (FST, see, e.g.,[5]), which we summarize briefly. A finite state transducer in a morphology processing application comprises

- an input alphabet Σ, usually letters or phones of the language of the morphology,

- an output alphabet Δ, usually composed of the input alphabet augmented by analysis tokens for the morphological description (like prefixes etc) and tokens for part-of-speech tags
- a finite set of states Q,
- a set of initial states $I \subset Q$,
- a set of final states $F \subset Q$,
- a finite set of transitions $E \subseteq Q \times \Sigma \times Q \times \Delta$.

An FST works by accepting input strings from Σ^* starting from an initial state through a path of transitions in E accepting one token from the input until reaching a final state in F – very much like a deterministic finite state automaton (DFA), however, each transition maps the input token consumed to a (possibly empty) output token in Δ^*. Using loose terminology, an FST can be said to translate a string from the upper language in Σ^* to the lower language in Δ^*. In effect, an accepted input string representing a natural language word is thus translated into an analysis string that usually contains segmented portions of the input word with additional analysis tokens.

A weighted FST has transition rules augmented by non-negative real valued weights

$$E \subseteq Q \times \Sigma \times (\mathbb{R}_+ \cup 0) \times Q \times \Delta$$

Weights on FST transitions are accumulated along paths using a tropical algebra, see [17].

Here is an (abridged) example of an output of the interactive morphology processor `hfst-lookup` (see [18]). This output was produed using a morphology for the German language (SMOR, see [23, 22]) rebuilt by the author for the HFST system [18]. For background on HFST-based morphology construction, including definition and construction of weighted morphologies, see [17]. Here is the (unweighted) output of the morphology processor given the input word `vorzeitig`.

```
vorzeitig       Vorzeit<NN>ig<SUFF><+ADJ><Pos><Adv>
vorzeitig       Vorzeit<NN>ig<SUFF><+ADJ><Pos><Pred>
vorzeitig       vor<PREF>Zeit<NN>ig<SUFF><+ADJ><Pos><Adv>
vorzeitig       vor<PREF>Zeit<NN>ig<SUFF><+ADJ><Pos><Pred>
vorzeitig       vor<PREF>zeitig<+ADJ><Pos><Adv>
vorzeitig       vor<PREF>zeitig<+ADJ><Pos><Pred>
```

It can be seen that ambiguities occur for the morphological as well as syntactical components of the analysis. In the analysis of running text (e.g., using `hfst-proc`, a tool provided by [18]), these ambiguities are resolved either using constraint propagation or Viterbi alignment on the weighted morphological analyses is involved. In the latter case, training of the analyzer on a large natural language corpus is needed such as to train not only local analysis probabilities but also part-of-speech tag transitions in word sequences.

Technically, a large morphology is constructed from the analyses provided by experts and / or by statistical analysis using the FST union and composition operations to give one single FST optimized for lookup performance, see [18].

The following examples demonstrate the capability of morphology processing in relation to compound word segmentation, here we omit the POS tags and weights for brevity.

Schuldnergruppenzuordnungen *Schuldner<NN>Gruppe<NN>zu<VPART>ordnen<V>ung<SUFF>*
Allgemeinverbindlichkeitsermahnungen *All<NN>gemein<ADJ>verbindlich<ADJ>keit<NN>*
<SUFF>er<VPREF>mahnen<V>ung<SUFF>

The principal benefits of applying morphological analysis to text corpora in preparation of computing a topic model are as follows–

1. the lemmatization (reduction to base morphems) leads to effectively lower vocabulary (or term set) size as it maps multiple inflected forms to a single base morphem,

2. the decomposition of compound words leads to effectively larger term counts as they accumulate for one token from several compounds (this effect cannot be produced by stemming).

The following section will present evaluations of those benefits for a real text corpus example.

Extensions to FST have been studied, in particular *pushdown transducers* that allow to process stack variables like a conventional pushdown automaton, however augmenting its functionality by a translation capability, see [5].

4 Using Morphologies for Document Preprocessing

In text mining based to the bag-of-words (BOW) approach, documents are usually preprocessed in several steps before specific text mining algorithms are applied. Typical steps include transformations to lower case, dropping punctuation and space or tab characters. A first non-trivial transformation is tokenization which in the simplest case establishes word boundaries, and in more complex cases, is used to identify multiword entities like person names, measure units (with preceding or following numerical values). Complex tokenizations have been one major contribution of research in named entity recognition (NER).

A second common and non-trivial transformation is stemming. Stemming results in token truncation in order to eliminate grammatical variations like case specific endings etc. Available open libraries like `R` text mining (`tm`, see [7]) include typically rule-based stemming packages like Porter stemming.

A serious limitation of stemming is the lack of deep linguistic processing which comes to bear especially upon processing inflection rich languages. In these, stemming will only recognize a very small fraction of forms of irregular verbs or irregular nouns. If the language of a corpus, in addition, makes rich use of compound words (as, e.g., in German corpora), stemming will also fail to decompose them and identify useful word components for the vocabulary. These capabilities are contributed to document preprocessing by morphological analysis.

4.1 Procedure for Morphological Processing of Text Corpora

The following pseudocode summarizes the procedure for morphological corpus analysis in a suitable way for enabling analytics related vocabulary sizes and coverage of document tokens by morphological lemmata.

For this purpose, first, all documents and their tokens are processed sequentially by a program combining tokenization with morphological analysis (in the case of the present research, `hfst-proc` [18] is used for the morphology in [23, 22]). The output of morphological analysis is a string for each token composed of a sequence of base lemmata together with their grammatical analysis and POS tags. As tokens usually are ambiguous permitting several analyses, the most probable such analysis is selected for further processing. Technically, the most probable morphological analysis can be computed from grammar and POS tags aligning a hidden Markov model as described earlier.

Second, the analysis string is stripped of grammatical and POS information to yield a single base lemma, or, in case of a compound word in the observed token, a sequence of pure base lemmata.

Third, this sequence (of length one in case of a non-compound word) is used to update a vocabulary list of lemmata together with their observed frequencies (created by document and integrated over documents in a final summary step).

Fourth, the vocabulary list is used to build an inverted index of observed tokens explained by a base lemma. For each lemma in the vocabulary, a list with tokens containing this lemma is maintained together with appropriate frequency counts. Note this list is formally a bag, as repetitions of tokens explained by a given lemma may occur. As in the preceding step, the inverted index is created for each document and a roll-up over documents is performed in a final summary module.

Finally, integration of the vocabularies, inverted indexes and frequencies is performed for all documents using a module containing usual aggregation operations (not detailed in the pseudocode). This aggregation step is currently implemented as a `gawk`[1] batch script operating on a collection of document vocabulary and index files.

The current implementation also does not compute a document-term frequency matrix as this is a standard function available from the `R tm` (text mining) package [7].

The implementation of these procedures is currently realized by a combination of GNU `gawk` scripts and UNIX shell scripts. `gawk` has built-in UTF-8 support and therefore does not require maintaining locale settings as would be needed, e.g., in C++. In addition, `gawk` has convenient pattern-based string splitting built-in functions and can readily be used for ragged arrays. These features make it a good choice for processing at least small up to medium sized text corpora.

[1] `https://www.gnu.org/software/gawk/`

```
input  : A corpus of text documents
output: A corpus of lemmatized documents, a lemma vocabulary and an inverted
        token index
foreach document i do
    foreach token j in i do
        get s:= morphologicalAnalysisString ( j ) ;
        lemmaList:= decompose(s, POStags) ;
        foreach lemma in lemmaList do
            if ! lemma in vocabularyList then
                appendTo (vocabularyList, entry(lemma, 1)) ;
                appendTo (inverseIndex, entry(lemma, token)) ;
            end
            else
                e=lookUp (vocabularyList, entry(lemma)) ;
                e.frequency++ ;
                if (inv=lookUp (inverseIndex,entry(lemma)) !=0 then
                    appendTo (inv.tokenlist, token) ;
                end
            end
        end
        appendTo (lemmatizedDoc, lemmaList) ;
    end
    update(globalVocab, vocabularyList) ;
    update(globalInvertedIndex, inverseIndex) ;
end
```

4.2 Effects of Morphological Processing on Vocabularies and Document Term Frequency Distributions

To investigate effects of morphological processing, several corpora are being examined at the LMU knowledge management research group. In this paper, results for a small corpus are presented. The corpus has been built from election program documents of four major democratic German parties published in the campaign for the general elections in Germany in 2013. The documents were segmented manually by chapters yielding an overall corpus size of 38 documents with overall around 200 000 tokens.

After applying stop word elimination the corpus was either stemmed or morphologically preprocessed. For stemming, the standard procedure available in `R` text mining package *tm* was used, [7]. For morphological analysis the tools and procedures described in the preceding section were used.

Morphological preprocessing reduces the size of the corpus vocabulary by aggregating tokens with common base lemmata into one observed term. For our test corpus, this effect is considerable, leading to a reduction in vocabulary size from 18372 to 5296. Additionally,

due to the decomposition of compound words, morphological preprocessing increases the number of the observed tokens, and, consequently, many term frequencies. Overall, the stemmed corpus has 83 180 tokens, while the morphologically preprocessed corpus has 100 771.

So, on the descriptive level of analysis, morphological processing leads to shift in term frequencies away from lower frequencies towards higher frequencies. This can be readily verified from the log-log plot of the corpus frequency table (plotting observed frequencies against frequencies of these frequencies), see fig. 2. E.g., for term frequency one indicating a term observed once in the entire corpus the stemmed data has 10 245 such terms, while the morphologically processed data has 1642 (not correcting for vocabulary size to keep the argument simple). On the other hand, the maximum observed frequency of a term in the stemmed data is 150 while it is 240 in the morphologically processed data.

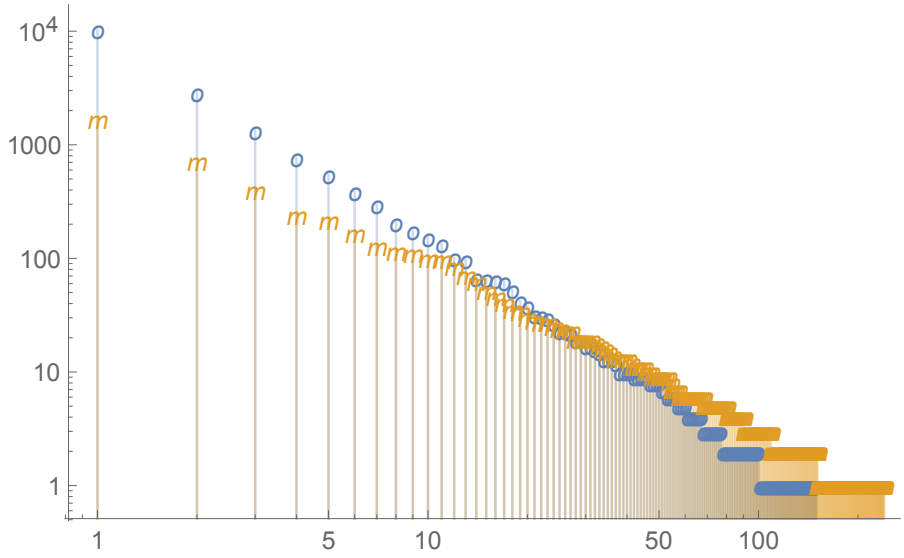
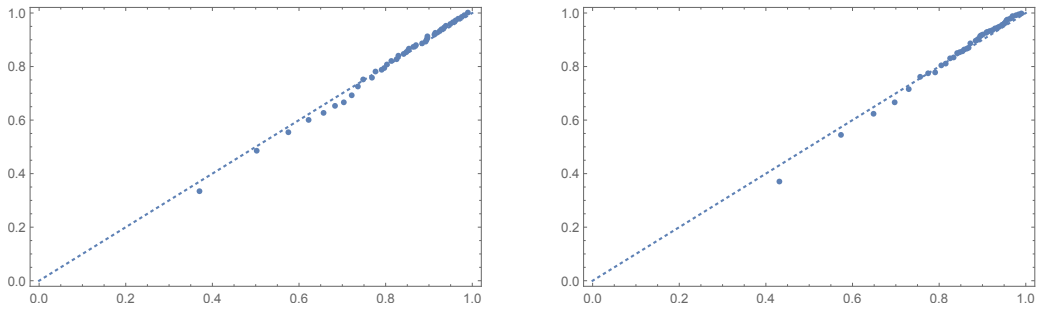


Fig. 2. Log-log plots of term frequency data from stemmed (marked o) vs. morphologically preprocessed (marked m) corpus. For details and explanation, see text.

Experiments with other corpora performed in our lab indicate similar effects. In order to grasp these effects in a statistically meaningful way, a Zipf distribution was fitted to stemmed vs morphologically preprocessed data. The Zipf distribution is a theoretical distribution of second order frequencies (or frequencies of term frequencies) in large text corpora that has been shown to fit natural language as well as similar data from other domains (see, e.g. [19]). A Zipf distribution would predict a straight line relating the log of first order frequencies to the log of second order frequencies.

In the case of the present corpus, first, the fit of an estimated Zipf distribution is very good for both data sets, as would be expected from the almost linear decreases in the plot in Fig. 2. It fact, the Pearson-Chi-Square fit test yielded .98 vs. .48 for the two data sets. The goodness of these fits can also be verified from the probability plots for the two

preprocessing procedures in Fig. 3.



(a) (b)

Fig. 3. Probability plots of term frequency data from stemmed 3(a) vs. morphologically preprocessed 3(b) corpus fitted to Zipf distributions. For detailed explanation, see text.

Second, it is interesting to see whether the distributions under both preprocessing procedures are different. This is the case with a p-value of 0.16 for the probability of the morphologically preprocessed data under the hypothesis of the stemmed data Zipf distribution parameter. On the other hand, a convincingly significant p-value of 0.04 is found for the probability of the stemmed preprocessed data under the hypothesis of the morphologically processed data Zipf distribution parameter. The difference in the p-values from either perspective can be attributed to the fact that the fitted parameter is higher for the morphologically processed data (.58) than for the other data set (.47), which leads to a wider spread distribution. We note that the Zipf distribution is a special case of the Riemann Zeta-distribution, and, as such, belongs to the long tail distributions. Specifically, in the case of the presently fitted parameters, both means and variances are infinite.

An additional instructive descriptive analysis can be performed based on the document term matrices (DTM). In information retrieval applications (see [19]), these matrices are usually analysed with singular-value decomposition (SVD) to capture latent semantic information (latent semantic analysis in the sense of [16]) hidden in term co-occurrences in documents.

As in LDA, SVD also leaves the analyst with the task to choose an appropriate reduced posterior dimensionality of the SVD-space. A useful interprestation of SVD in case of document term matrix inputs is that of a best rank n approximation B of a matrix A that minimizes the Frobenius norm of $||B-A||$ where B is taken to be the SVD of A. As shown in [14], the remaining part of $||B-A||$ after taking into account k singular values E_k is.

$$E_k = \sqrt{\frac{\sum_{i=k}^{l} \sigma_i^2}{\sum_{i=1}^{l} \sigma_i^2}}$$

Fig. 4 shows the sequences of residual errors in the sense just defined for the SVD decompositions of document term matrices from our election campaign corpus with stemming vs morphological processing. It is readily verified that morphological processing allows to capture the overall data variability with a considerably lower number of dimensions for

any given error level. E.g., to capture ca. 80% of the variability, a 20-dimensional space would suffice for morphological processed data while 25 dimensions would be needed for stemmed data.

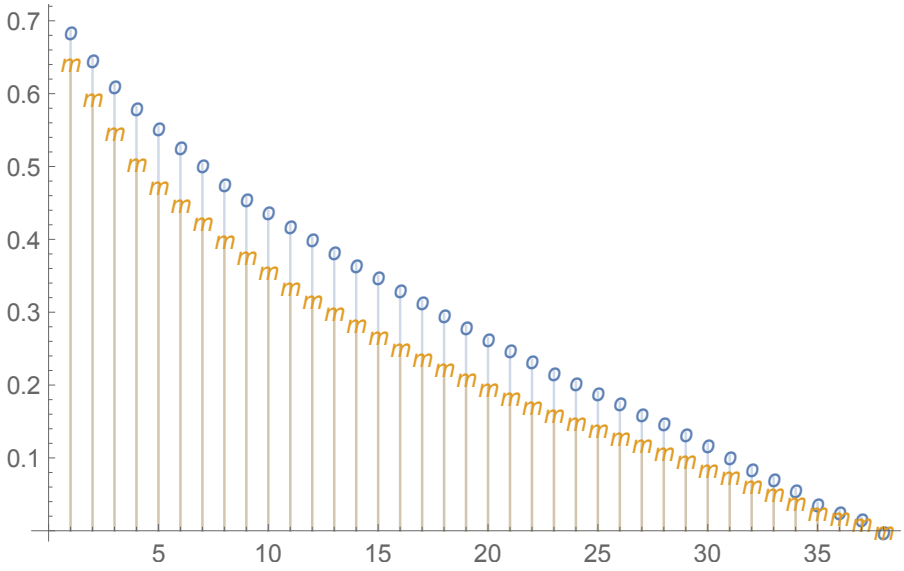


Fig. 4. Sequences of residual errors for SVD decompositions of document term matrices from the election campaign corpus with stemming (marked o) vs morphological processing (marked m) For details and explanation, see text.

Finally, the impact of morphological preprocessing on topic model inference can be assessed. Here is a summary of initial findings for the election program corpus described above.

In some experiments with HDP (using the recent implementation [4] with a few minor amendments), the advantages of the analyses by LDA with morphological preprocessed corpora could be confirmed.

In addition, HDP with morphological preprocessed corpora leads to substantially lower estimates of topic numbers. For our example, the election campaign corpus, running HDP with default parameter settings for 1000 iterations for the morphologically preprocessed corpus leads to an estimation of 67 topics with an overall estimated log-likelihood of -772600.402 and an average document log-likelihood of -7.66689. Under the same conditions, the stemmed original corpus leads to a model with 80 topics and an overall estimated log-likelihood of -794623.148 with a per document average log-likelihood of -9.55306. Thus, morphological preprocessing allowed a decrease in the number of estimated topics by ca. 16%, and a considerable improvement in posterior document log-likelihood of about 20%.

As for Latent Dirichlet Allocation (LDA), the topic numbers found by HDP (67 vs. 80) were used to compute LDA models for the corpus in stemmed vs morphologically preprocessed versions. For the exploratory purpose of the present paper, LDA was run

from R topicmodels [12] using the variational inference algorithm as detailed in [3] with the default settings (start value of the prior document-topic Dirichlet dispersion parameter of .625, tolerance 10^{-6} and 1000 iterations for variaional inference, tolerance 10^{-4} and 500 iterations for the document expectation step of the EM part of the variational algorithm). In order to see potential benefits of other fixed LDA topic numbers, additional analyses were run for topic numbers 19, 38 (half and full corpus size). As can be verified from Fig. 5, for all choices of topic numbers, the posterior perplexity value of the models based on the morphologically preprocessed corpus outperform those for the stemmed corpus (perplexity, here, is derived from posterior document likelihoods normalized by the number of tokens per document, see section 7.1 in [3]).

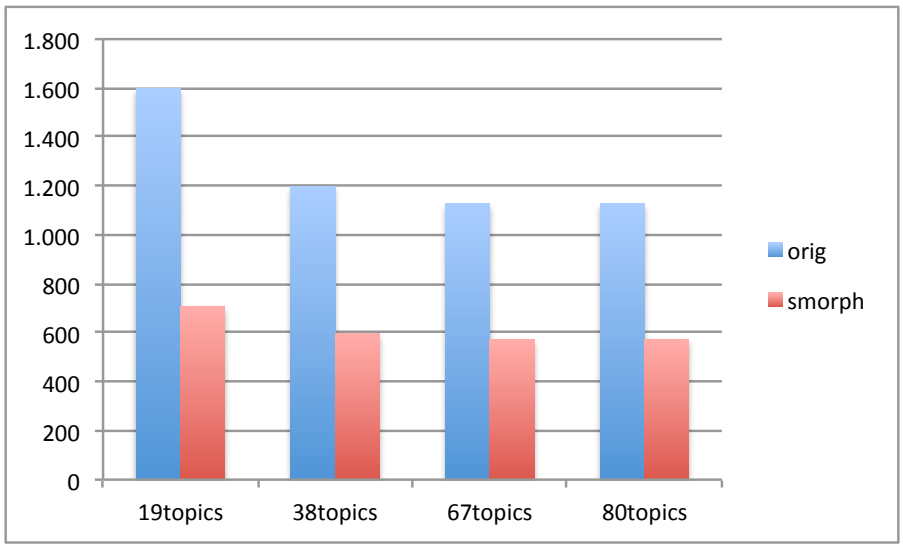


Fig. 5. Perplexities of LDA models estimated for various topic numbers comparing morphologically preprocessed (legended smorph) to stemmed versions of the election campaign text corpus (legended orig). (Lower values indicate better model fit.) For details and explanation, see text.

To sum up, initial simulations of topic models built on morphologically preprocessed corpora show substantially improved model fits compared to simple preprocessing by stemming or related procedures.

5 Conclusion

In the present paper, we reviewed some essential features of the Hierarchical Dirichlet Process (HDP), a non-parametric topic modelling approach that extends Latent Dirichlet allocation (LDA) to allow for a dynamically assigned number of topics during the model estimation phase for a given corpus. (Recall that non-parametric here only means that the number of model parameters is estimation dependent.)

From a pragmatic point of view, HDP alleviates the burden of topic number testing from content analysts applying topic modelling. A significant problem involved with this testing

is that computations of LDA models for different topic numbers are independent in the sense that there is no topic identity or modification across different runs of an LDA estimator. Together with the proven non-uniqueness of LDA solutions this means that in principle a sample of models for each topic number would need to be computed and assessed in order to yield a reasonable guideline for choosing an appropriate topic number. HDP does compute a sequence of partial topic models such that increases or decreases in the number of estimated topics vary on a common base model. Even if an appropriate use of this technique also requires several simulation runs and parameter adjustments the resulting topic number proposals are in general similar and can readily be used by content analysts. This has been found already in a thesis from our work group at LMU Munich for a large document corpus of an education institution [21]. The findings for the election campaign corpus confirm these results, moreover, they highlight the importance of morphological preprocessing for ensuring high model quality.

Related to future work, the simulations performed are planned to be extended by some fine tuning of the estimation procedures for both LDA and HDP. Initial results show that fine tuning parameters and simulation procedures can improve topic model quality substantially. It is also planned to further investigate morphological processing in combination with extended linguistic features in its effects on corpus descriptors like term frequencies.

References

[1] David M. Blei. *lda-c*. http://www.cs.princeton.edu1 blei/lda-c/, 2003. URL: http://www.cs.princeton.edu1%20blei/lda-c/.

[2] David M. Blei, Thomas L. Griffiths, and Michael I. Jordan. *The nested Chinese restaurant process and hierarchical topic models*. 2007. eprint: 0710.0845. URL: http://arxiv.org/abs/0710.0845.

[3] David M. Blei, Andrew Ng, and Michael Jordan. "Latent Dirichlet allocation". In: *JMLR* 3 (Jan. 2003), pp. 993–1022.

[4] David M. Blei and Chong Wang. *Hierarchical Dirichlet Process (with Split-Merge Operations)*. 2013. URL: https://github.com/blei-lab/hdp.

[5] C. Choffrut and K. Culik II. "Properties of Finite and Pushdown Transducers". In: *SIAM Journal on Computing* 12.2 (May 1, 1983), pp. 300–315. DOI: 10.1137/0212019. URL: http://dx.doi.org/10.1137/0212019.

[6] Laura Dietz. *Directed Factor Graph Notation for Generative Models*. 2011. URL: https://github.com/jluttine/tikz-bayesnet.

[7] Ingo Feinerer. "An introduction to the tm package – Text mining in R". In: *R News* 8.2 (2012), pp. 19–22.

[8] Thomas Ferguson. "Bayesian Analysis of some nonparametric problems". In: *The Annals of Statistics* 1.2 (1973), pp. 209–230.

[9] Bela A. Frigyik, Amol Kapila, and Maya R. Gupta. *Introduction to the Dirichlet Distribution and Related Processes*. Tech. rep. University of Washington, Dpt. Electrical Engineering, 2010.

[10] Lise Getoor and Ben Taskar, eds. *Introduction to Statistical Relational Learning*. Cambridge, MA: MIT Press, 2007.

[11] Thomas L. Griffiths and Mark Steyvers. "Finding Scientific Topics". In: *Proceedings of the National Academy of Sciences of the United States of America* 101.suppl. 1 (Apr. 2004), pp. 5228–5235.

[12] Bettina Grün and Kurt Hornik. "topicmodels: An R package for fitting topic models". In: 40 (2012), pp. 1–30. URL: http://www.jstatsoft.org/v40/i13/.

[13] David Heckerman, Chris Meck, and Daphne Koller. "Probabilistic Entity-Relationship Models, PRMs, and Plate Models". In: *Introduction to Statistical Relational Learning*. Ed. by Lise Getoor and Ben Taskar. Cambridge, MA: MIT Press, 2007, pp. 201–238.

[14] Dan Kalman. *A Singularly Valuable Decomposition: The SVD of a Matrix*. 2002. URL: http://www.math.umn.edu/~lerman/math5467/svd.pdf.

[15] Frank Kschischang, Brendan J. Frey, and Hans-Andrea Loeliger. "Factor Graphs and the Sum-Product Algorithm". In: *IEEE Transactions on Information Theory* 47.2 (2001), pp. 498–519.

[16] Thomas K. Landauer and Susan T. Dumais. "A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge". In: *Psychological Review* 104 (1997), pp. 211–240. URL: 10.1037//0033-295X.104.2.211.

[17] Krister Linden and Tommi Pirinen. *Weighting finite-state morphological analyzers using hfst tools*. URL: http://www.researchgate.net/publication/228912097.

[18] Krister Linden et al. *HFST–a System for Creating NLP Tools*. Tech. rep. University of Helsinki, Dpt. of Modern Languages, 2011.

[19] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. "An Introduction to Information Retrieval". In: (2009). URL: http://www.informationretrieval.org/.

[20] Thomas P. Minka and John Winn. *Gates: A Graphical Notation for Mixture Models*. Tech. rep. MSR-TR-2005-173. Microsoft Research, 2005.

[21] Carina Müller. "Identifizieren gemeinsamer inhaltlicher Kriterien in einem heterogenen kirchlichen Bildungsangebot mithilfe von Topic Modelling-Ansätzen". Master Thesis. LMU University of Munich, Chair of Knowledge Management, 2015.

[22] Helmut Schmid. *A Programming Lanaguage for Finite State Transducers*. URL: http://www.cis.uni-muenchen.de/~schmid/tools/SFST/.

[23] Helmut Schmid. *Stuttgart Morphology for German SMOR*. URL: http://www.cis.uni-muenchen.de/~schmid/tools/SMOR/.

[24] Jayaram Sethuraman. "A constructive definition of Dirichlet priors". In: *Statistica Sinica* 4 (1994), pp. 639–650.

[25] Marcus Spies and Monika Jungemann-Dorner. "Big Textual Data Analytics and Knowledge Management". In: *Big Data Computing*. Ed. by Rajendra Akerkar. Chapman and Hall/CRC, 2013. Chap. 23, pp. 501–537. ISBN: 978-1-4665-7837. DOI: doi:10.1201/b16014-23. URL: http://dx.doi.org/10.1201/b16014-1.

[26] Mark Steyvers and Tom Griffiths. *Matlab Topic Modeling Toolbox*. 2005. URL: http://psiexp.ss.uci.edu/research/programs%7B%5C_%7Ddata/toolbox.htm.

[27] Charles Sutton and Andrew McCallum. "An Introduction to Conditional Random Fields for Relational Learning". In: *Introduction to Statistical Relational Learning*. Ed. by Lise Getoor and Ben Taskar. Cambridge, MA: MIT Press, 2007.

[28] Yee Whye Teh et al. "Hierarchical Dirichlet Processes". In: *JASA* 101.476 (2006), pp. 1566–1581. ISSN: 0162-1459. URL: http://dx.doi.org/10.1198/016214506000000302.

[29] Konstantin Vorontsov and Anna Potapenko. "Tutorial on Probabilistic Topic Modeling: Additive Regularization for Stochastic Matrix Factorization". In: *Analysis of Images, Social Networks and Texts*. Ed. by Mikhail Yu Khachay et al. Vol. 436. Communications in Computer and Information Science 3. Springer International, Switzerland, 2014, pp. 29–46.

[30] Limin Yao, David Mimno, and Andrew McCallum. "Efficient Methods for Topic Model Inference on Streaming Document Collections". In: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '09. Paris, France: ACM, 2009, pp. 937–946. ISBN: 978-1-60558-495-9. DOI: 10.1145/1557019.1557121. URL: http://doi.acm.org/10.1145/1557019.1557121.