DEGREE PROJECT IN COMPUTER SCIENCE AND ENGINEERING,
FIRST CYCLE, 15 CREDITS
STOCKHOLM, SWEDEN 2016

Finding competitors using Latent Dirichlet Allocation

ARNEKVIST, ISAC

ERICSON, LUDVIG

KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF COMPUTER SCIENCE AND COMMUNICATION

Finding competitors using Latent Dirichlet Allocation

Hitta konkurrenter med Latent Dirichlet Allocation

Supervisor: Jens Lagergren

Abstract

Identifying business competitors is of interest to many, but is becoming increasingly hard in an expanding global market. The aim of this report is to investigate whether Latent Dirichlet Allocation (LDA) can be used to identify and rank competitors based on distances between LDA representations of company descriptions. The performance of the LDA model was compared to that of bag-of-words and random ordering by evaluating then comparing them on a handful of common information retrieval metrics. Several different distance metrics were evaluated to determine which metric had best correspondence between representation distance and companies being competitors. Cosine similarity was found to outperform the other distance metrics. While both LDA and bag-of-words representations were found to be significantly better than random ordering, LDA was found to perform worse than bag-of-words. However, computation of distance metrics was considerably faster for LDA representations. The LDA representations capture features that are not helpful for identifying competitors, and it is suggested that LDA representations could be used together with some other data source or heuristic.

Sammanfattning

Det finns ett intresse av att kunna identifiera affärskonkurrenter, men detta blir allt svårare på en ständigt växande och alltmer global marknad. Syftet med denna rapport är att undersöka om Latent Dirichlet Allocation (LDA) kan användas för att identifiera och rangordna konkurrenter. Detta genom att jämföra avstånden mellan LDA-representationerna av dessas företagsbeskrivningar. Effektiviteten av LDA i detta syfte jämfördes med den för bag-of-words samt slumpmässig ordning, detta med hjälp av några vanliga informationsteoretiska mått. Flera olika avståndsmått utvärderades för att bestämma vilken av dessa som bäst åstadkommer att konkurrerande företag hamnar nära varandra. I detta fall fanns Cosine similarity överträffa andra avståndsmått. Medan både LDA och bag-of-words konstaterades vara signifikant bättre än slumpmässig ordning så fanns att LDA presterar kvalitativt sämre än bag-of-words. Uträkning av avståndsmått var dock betydligt snabbare med LDA-representationer. Att omvandla webbinnehåll till LDA-representationer fångar dock vissa ospecifika likheter som inte nödvändigt beskriver konkurrenter. Det kan möjligen vara fördelaktigt att använda LDA-representationer ihop med någon ytterligare datakälla och/eller heuristik.

Contents

1 Introduction

> *If you know the enemy and know yourself, you need not fear the result of a hundred battles. If you know yourself but not the enemy, for every victory gained you will also suffer a defeat. If you know neither the enemy nor yourself, you will succumb in every battle.*
>
> – 孫子 (Sun Tzu)

Identifying competitors is a vital part to any successful business strategy, and this is becoming increasingly hard in an ever-growing global market. There are many potential users of such information, e.g. business owners assessing their competition, consumers comparing their options, or a finance company making a market valuation.

Text analysis is a relatively well-researched field, and in particular there is a rich variety of methods for performing text similarity ranking. *Topic models* provide one way of representing documents as a probability distribution over topics. Distance metrics can then be applied on the distributions to rank documents on distance/non-similarity.

Latent Dirichlet allocation (LDA) is one of the simplest topic models and one on which many others are based. The aim of this report is to evaluate the performance of LDA combined with various distance measures for identifying competitors by similarities of the businesses' descriptions in a third-party database and the text on the their web pages.

1.1 Problem Statement

Can the topic model LDA be used to identify competitors given businesses' textual descriptions and web page contents?

2 Background

High-dimensional data representations, such as word frequencies in text documents, are problematic. This section first introduces some background to why this is, along with some means of dealing with such issues, and especially how LDA can be used to this end for the case of text documents. Secondly, some evaluation methods for the method are introduced, and lastly some related work is discussed.

2.1 Latent space models

In a latent space model one makes the assumption that observed data is generated by some process with a lower *intrinsic dimensionality* than the observed representation. To clarify what this means, consider a 2-megapixel image, where each pixel has three color channels – that is in total 6 million dimensions. A vanishingly small subset of this high-dimensional space represent actual images. And more, what the picture actually represents might be "A tennis ball with size w at coordinates x, y". So this

high dimensional space can actually be reduced and described using only three dimensions/variables. These three dimensions are what is called *latent variables* and the method of finding them and reducing the dimensionality is simply called *dimensionality reduction*.

This high dimensionality is a recurring problem often called *the curse of dimensionality* – representations necessarily fall far apart in this high-dimensional space, and it becomes harder to group the data in any meaningful way. Latent space models can solve this problem if the number of latent variables is low, and can thus be viewed as a type of dimensionality reduction (Bishop, 2006).

A common representation of text documents is *bag-of-words*, which is quite simply a count of the number of each word in a document (James et al., 2014). This representation therefore becomes as high-dimensional as the vocabulary used, which is usually far too high. Here, for the same reason, latent space models are often used.

2.2 Topic models

There are several ways to reduce the dimensionality, but still try to to keep the significant part of the data. One of them is to imagine that a document is generated from a distribution of topics. For instance, perhaps an article about FIFA has the topics corruption and sports, while an article on Google is about IT, research and machine learning. The representation can now be a vector with as many elements as there are topics in the model, each component describes to what extent the corresponding topic is responsible for the document content. These are called *topic models* (Blei et al., 2003). Topic models are useful if the number of distinct words in the text is high, since topic models can be used to reduce the dimensionality to the number of topics.

2.3 Categorical & Dirichlet distributions

The topics are a discrete number of possible outcomes usually just numbered 1 to K. Inducing a probability distribution over these topics gives rise to a categorical distribution, here parameterized by $\boldsymbol{\theta} = (\theta_1, \cdots, \theta_K)$. Let X be a stochastic variable representing a latent topic, where $X \in \{1, ..., K\}$, then the probability of some outcome i is θ_i, in other words

$$\mathrm{p}(X = i|\boldsymbol{\theta}) = \theta_i \equiv X \sim \mathrm{Cat}(\boldsymbol{\theta}) \tag{1}$$

Not all parameter choices are equally likely, or desirable. In the case of topic distributions for a document, a uniform distribution is equivalent to saying the documents can be described by an equal mixture of all topics. Clearly, we desire to to encode our belief, or our preference, in non-uniform distributions, and so introduce a *prior distribution* over the parameters $\boldsymbol{\theta}$. Due to conjugacy, there is only one form for the prior probability here, namely a Dirichlet distribution. Dirichlet distributions are of the form

$$\mathrm{p}(\boldsymbol{\theta}|\boldsymbol{\alpha}) = \frac{1}{B(\boldsymbol{\alpha})} \prod_{i=1}^{K} \theta_i^{\alpha_i - 1} \equiv \boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha}) \tag{2}$$

where the concentration $\boldsymbol{\alpha}$ describes how much probability mass to assign to uniform versus non-uniform categorical distributions. A large α_j means that small θ_j's are less likely, since $\theta_j^{\alpha_j - 1}$ comes closer to zero for small values of θ_j. On the other hand, a small value for some α_j increases the probability of smaller θ_j compared to for larger α_j. $B(\boldsymbol{\alpha})$ is a normalizing function, scaling the probability density function such that its integral becomes one.

Dirichlet distributions are as previously noted common in models where latent variables are assumed to be finite and discrete. This is due to the fact that the Dirichlet distribution is the only *conjugate distribution* to the categorical distribution.

2.3.1 Parameter choices for the Dirichlet distribution

In our model, i.e. there is a mixture of topics for every document in a set of documents, we prefer topic distributions that are non-uniform. The aim is to strike a good balance: assigning only one topic to a document seems too simple, and the opposite seems vague and non-informative. This behavior is dictated by the concentration *hyperparameter* for the model, and needs be adjusted and optimized to be pertinent to the application at hand.

2.4 Latent Dirichlet Allocation

With the concepts developed so far, we are now ready to look at *LDA*, or *Latent Dirichlet Allocation*, which is the topic model to be used in this report. In LDA the assumption is that the document is created from a number of topics with different probability. For each topic, there is a probability distribution that each word belongs to the topic, a topic-word distribution.

LDA can be thought of as a generative model, because it's possible to sample from the distributions and generate new documents.

One weakness in LDA is that it doesn't model the order of the words in a document, and so does not understand grammatical constructs other than by mere simultaneous occurrence of words (Blei et al., 2003).

2.4.1 The generative process

The graphical model underlying LDA is shown in fig. 1. For each topic k of the K topics, a topic-word distribution $\text{Cat}(\boldsymbol{\beta}_k)$ is sampled from the prior $\text{Dir}(\boldsymbol{\eta})$ with N possible outcomes; and correspondingly, for each document d of the D documents, a topic distribution $\text{Cat}(\boldsymbol{\theta}_d)$ is sampled from the prior $\text{Dir}(\boldsymbol{\alpha})$.

Often, there is no reason to suspect that one topic will be more "concentrated" than any other, so the parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\eta}$ are often set to scalars values α and η.

$\text{Cat}(\boldsymbol{\theta}_d)$ is thus the distribution of topics for a document d, from which we can sample some number of words. For every word, we first need to sample a topic k from $\text{Cat}(\boldsymbol{\theta}_d)$, then we can sample a word from the topic-word distribution $\text{Cat}(\boldsymbol{\beta}_k)$.

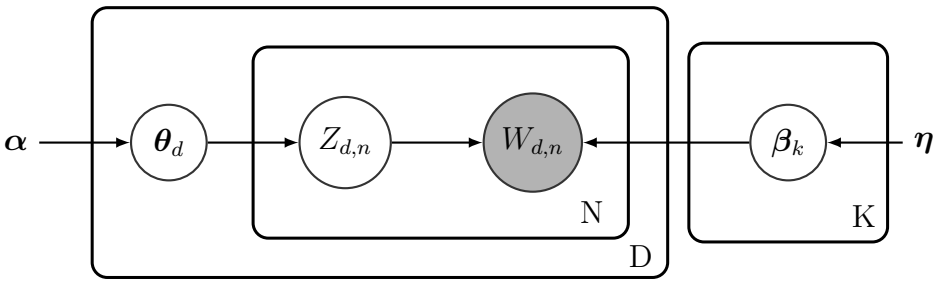


Figure 1: The graphical model of LDA represented with *plate notation*. Each plate consists of a set of variables that are conditionally independent of the other plates. Here N is the number of words in the vocabulary, D is the number of documents, K is the number of topics, $W_{d,n}$ is the nth observed word in the dth document, $Z_{d,n}$ is the latent topic for the nth word in the dth document, $\boldsymbol{\theta}_d$ is the dth document's topic distribution, $\boldsymbol{\beta}_k$ is the kth topic's word distribution, with $\boldsymbol{\theta}_d \sim \text{Dir}(\boldsymbol{\alpha})$ and $\boldsymbol{\beta}_k \sim \text{Dir}(\boldsymbol{\eta})$.

In LDA, the words $W_{d,n}$ are observed variables (marked gray in the figure), therefore when learning the $\text{Cat}(\boldsymbol{\theta}_d)$ and $\text{Cat}(\boldsymbol{\beta}_k)$ of such a model, one has to "go backwards" towards the generative process. One well-established method of doing so is *expectation maximization* (Blei and Lafferty, 2009).

2.5 Evaluation

The quality of a system that returns its results ranked in order of relevance, notably one that finds the closest competitors to some business, can be assessed in multiple ways. Many are extensions or adaptations of more primitive measures, and are presented in that order below. Given some query for a document A, other documents then have a binary label with respect to A, relevant or non-relevant, and these form the set of relevant documents R given A. The result set is an ordered set where high scores for the metrics below means that relevant documents are listed early in this set (Manning et al., 2008).

The first metric is *Precision* and is defined given all relevant documents R for a query and the result set S as,

$$\text{P} = \frac{|R \cap S|}{|S|}, \tag{3}$$

i.e. the proportion of the number of relevant documents in the result set. Then there is also *Recall* which is a similar measure, but instead measures the proportion of the relevant documents found in the result:

$$\text{Rec} = \frac{|R \cap S|}{|R|}. \tag{4}$$

Notably, recall is independent of the result set length. An empty result implies no

recall, and the result being the entire document set implies full recall. Precision is further used in the metric *P@N*, or precision at N, which is the precision as defined above for the top N results.

Using both precision and recall, *Average Precision* can be defined with the purpose of being a measure of whether the result has the relevant documents at the top. This is defined:

$$\text{AveP} = \sum_{k=1}^{N} \text{p}(k)\Delta r(k) \tag{5}$$

where k is an index variable from the topmost to the lowest result. The last factor is the difference in recall from the previous result, that is,

$$\begin{aligned} \Delta r(1) &= \text{Rec}(1) \\ \Delta r(k) &= \text{Rec}(k) - \text{Rec}(k-1). \end{aligned} \tag{6}$$

Moreover we have *MAP* which is short for *Mean Average Precision*, in which a set of queries Q are made, then the average precision $\text{AveP}(Q_i)$ is calculated for each query. MAP is then simply the mean of these averages:

$$\text{MAP} = \sum_{i=1}^{|Q|} \frac{\text{AveP}(Q_i)}{|Q|} \tag{7}$$

Lastly *NDCG* or *Normalized Discounted Cumulative Gain* is a means of scoring the ordered result set given some relevance score $r_i \in \{0, 1, ...\}$ for each document i. Higher scores equal more relevant. Let us first define DCG_p, that is, the discontinued cumulative gain for a result set of length p:

$$\text{DCG}_p = \sum_{i=1}^{p} \frac{2^{r_i} - 1}{\log_2(i+1)} \tag{8}$$

The normalized DCG is done by dividing by the *ideal* DCG, the IDCG, which is the DCG had the result set been optimally ordered – that is, in order of greatest relevance score first:

$$\text{NDCG}_p = \frac{\text{DCG}_p}{\text{IDCG}_p} \tag{9}$$

2.6 Related research

Extensive searches on *Google Scholar* shows that areas such as latent-space models, topic models, information retrieval and LDA are well-known and researched. However, few studies have been done on using topic models on the competitor analysis problem.

2.6.1 Analysis of known competitors

Leong et al. (2004) have in a study used text analysis for the purpose of analysing competitors. Instead of automatically finding competitors they have aimed to analyze already-known competitors. The analysis was like LDA based on finding central themes,

but the underlying model wasn't described other than that some pre-existing software was used.

2.6.2 PatentMiner

Another attempt at automatically finding competitors is due to Tang et al. (2012), by searching through patents and analysing them with topic models combined with graph of the authors, companies and patents. They used three different metrics to determine the rank of competitors c' to some business c:

- word-based similarity, in which cosine similarity was applied to a bag-of-words representation of each result;
- topic-based similarity, in which Kullback-Leibler divergence was used; and,
- probability-based similarity, where latent topic variables $\{z_i\}_{i=1}^{K}$ from LDA were used to calculate a ranking score $S(c, c') \propto \sum_{z_i} \mathrm{p}(z_i|c)\,\mathrm{p}(z_i|c')$.

The results were then compared to some existing ground truth labels using *P@N*, *MAP* and *NDCG*.

3 Method

3.1 Text and competitor mining

The corpus was extracted from the online database CrunchBase, in particular the descriptions and web page addresses were extracted. The web pages were then crawled from a server in the United States to avoid undesirable automated localization of the web pages, which is often based on the client's IP address. The resulting HTML documents were then scrubbed using the Python package *BeautifulSoup*, and combined with the textual description in the database.

The CrunchBase database contains information on known competitors, albeit few. This was used to form an undirected graph, where each pair of competitors is an edge in the graph. The corpus was then partitioned into three sets: training, validation and test sets. The training set were those companies in the database that had no known competitors. The validation and test sets were formed by splitting the aforementioned competitor graph in twain, while preserving as many edges as possible. This was done because both validation and test procedures need labelled data, which was a scarce to be found: out of 180 000 companies, only about 20 000 had any known competitors. This is further detailed in section 3.6.

3.2 Vectorization parameters

Two different text vectorizers from the Python package *scikit-learn* were used: the first simply counts the occurrences of each word in each document, while the other uses

a weighting scheme known as *term frequency–inverse document frequency* (*tf-idf*). The vectorizers also have two cut-off parameters: maximum document frequency (*max-df*), and minimum document frequency (*min-df*), which is a minimum and maximum ratio of documents that a word is allowed to occur in before being removed from the learned vocabulary. These parameters were optimized by sampling *min-df* uniformly from 2 to 100, and using *tf-idf* weighting with a 20% probability. See section 3.5 for further details.

3.3 LDA

Scikit-learn's LDA was used to find the latent representations. The hyperparameter in this stage to consider is the number of topics. This was sampled uniformly between 20 and 400. Other priors was left unchanged from the default values.

3.4 Model evaluation

Given a query, the documents in the corpus were ranked according to some distance or similarity measure on the latent representations. The following measures were used:

- Squared Euclidean distance, $d_{\mathrm{L}_2} = ||\mathbf{u} - \mathbf{v}||_2$
- Cosine similarity, $d_{\cos} = 1 - \frac{\mathbf{u}\cdot\mathbf{v}}{\sqrt{\mathbf{u}\cdot\mathbf{u}\times\mathbf{v}\cdot\mathbf{v}}}$
- Probability-based similarity, $d_{\text{prob}} = 1 - \mathbf{u}\cdot\mathbf{v}$
- Kullback-Leibler divergence, $d_{\mathrm{KL}} = \sum_i u_i \ln \frac{u_i}{v_i}$
- Jensen-Shannon distance, $d_{\text{Jens}} = \frac{d_{\mathrm{KL}}(\mathbf{u},\mathbf{v})+d_{\mathrm{KL}}(\mathbf{v},\mathbf{u})}{2}$

To be able to compare the effectiveness of latent topic representations on this kind of task, the bag-of-words representations were also used where the degree of similarity was calculated using:

- Squared Euclidean distance
- Cosine similarity

Both the queries on the latent representations and the bag-of-words were evaluated using *P@N*, *MAP* and *NDCG*. Consider companies as the nodes of a graph and non-directed edges representing competitors. Given a query for some company, its neighbors are considered relevant and all other companies non-relevant when calculating *P@N* and *MAP*. For *NDCG*, neighbors of the queried company were given relevance score k, neighbors of neighbors $k-1$ and so on and companies k steps and further away in the graph were given score 0. The scores were found using breadth-first search. Several random queries were made for all similarity measures and the evaluation metrics were averaged.

3.5 Hyperparameter selection

Several runs of the vectorization and LDA modelling were made with randomly selected parameters on a validation set to find the optimal hyperparameters. The method of randomly choosing parameters is usually referred to as a random grid search (Bergstra and Bengio, 2012). Each of these runs were then evaluated using the metrics *P@N*, *MAP* and *NDCG*. Each run was evaluated with enough queries that the relative standard error of the mean (SEM) for the score was below some ad hoc threshold. The threshold was chosen so that the evaluation would terminate within a reasonbale time frame. In the process of selecting parameters, the distance measure producing the highest scores was first chosen, one for the bag-of-words queries and one for the LDA queries. Then given that metric, the parameters for the runs with highest scores were chosen to train the final test model to evaluate on the test set. Due to long running times in grid search, a smaller subset of the training data was used in this stage compared to when training the final LDA model.

3.6 Minimum bisection problem

When analysing the performance of a statistical method, and in particular methods based on making inferences from observations, it's not to "train on the test data," as it would be trivial to design a method that would have a 100% test score if it a-priori got to know the test data (Bishop, 2006). Splitting data into training and tests sets is therefore completely necessary.

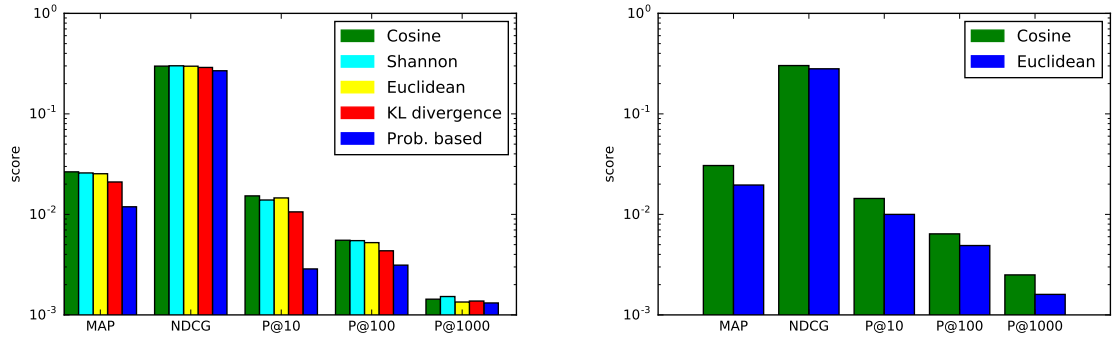
The dataset used in this report is a graph, and as such, there was a question on how to best cut a graph into two approximately equal parts while removing as few edges as possible. Each company in the graph must have at least one competitor for the information retrieval metrics to be defined. This means that when cutting the graph into two parts, several companies may end up with no competitors if the only edge connecting them to another company was deleted. So, when splitting the graph, a lot of data can be lost if the cut was unlucky. This optimisation problem is well-known in the literature, and it's a special case of the graph partitioning problem often called the *minimum bisection problem*. The problem is NP-complete, meaning no known algorithm can solve it in polynomial time; hence the only viable option is to use an approximation method, or a heuristic approach. (Garey and Johnson, 1979).

We evaluated some promising methods from current literature, but found that most methods are either impractical to apply due to the size of our dataset, or existing implementations were not readily available. This includes spectral modularity-based methods, stochastic block models, and Kernighan-Lin decomposition.

A straightforward simulated annealing-based algorithm was the strategy that yielded the most satisfactory results. We used the cooling schedule

$$T = \frac{T_{min}^{t}}{T_{max}^{t-1}} - T_{min}.$$

The key to achieving satisfying results with this algorithm was to minimize the amount of copying done; instead of copying the Markov chain, an undo-based algorithm was



(a) Side-by-side score comparisons of the different distance measures on LDA representations.

(b) Side-by-side score comparisons of the different distance measures on bag-of-words.

Figure 2: Comparison of the different distance metrics in both LDA and bag-of-words models.

Table 1: Chosen parameters for training test model

	Bag-of-words	LDA
No. of topics	–	300
min-df	6	50
tf-idf	No	No

employed. With each step, a function is produced that can undo the step taken, as opposed to relying on copying the entire state. In this way, the performance drastically improved, providing a better graph bisection in less time.

4 Results

4.1 Parameter choices

In figure 2(a) and 2(b), the different evaluation scores for all of the distance measures given topic and bag-of-word representations are shown. These are averages over several runs and several parameters. The cosine metric was chosen as distance metric for the LDA representations.

Looking at the parameters and their respective scores given cosine as a distance metric, parameters with the best score were chosen to train the final model on. These are listed in table 1.

4.2 Quantitative observations

The final model was trained on the parameters chosen in the validation step, then evaluated. The results can be seen in figures 3 and 4, plotted as the mean score over several

runs. Due to the central limit theorem, the mean score has asymptotically Gaussian distribution, and the confidence intervals are based on this approximation, using mean standard error. Bag-of-words performed significantly better than LDA when evaluated with MAP and P@10. The mean of metrics P@100 and NDCG were also better for bag-of-words, although not significant. Both methods, using any metric, performed significantly better than random query results. Since LDA is a non-deterministic method, 10 LDA models were trained to ascertain that the results were consistent, which they were.

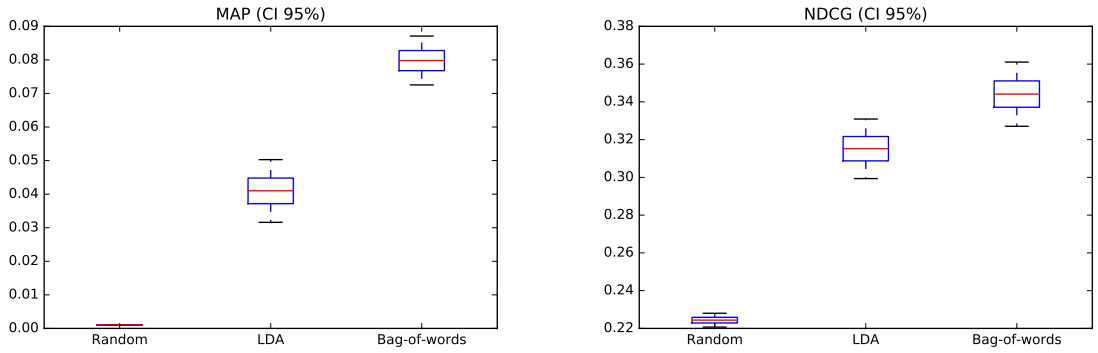


Figure 3: Box plots of final results using MAP and NDCG as metrics, confidence interval (CI) is 95%

4.3 Qualitative observations

4.3.1 Topics found

In table 2, some of the topics found are listed. Some are what might be considered more intuitively close to how anyone would describe what a topic is, that is, describing things such as travel, technology, or fitness. Other topics are just the most common words in a certain language or common elements such as numbers. When looking at a such a large amount of web pages, mock-up words also appear as its own topic.

4.3.2 Web-specific content

Some results were found qualitatively to be incorrect. Closer examination revealed this to be due to topics consisting of non-semantic information, such as commonalities to some types of web pages.

Example: stardoll.com vs talenthouse.com The two companies were close in topic space, and the most common words in the topics they have in common are listed in table 3. Both pages had a menu for selecting languages, see fig. 5. An interesting aside is that the names of the languages are not the same in these two pages, but both are

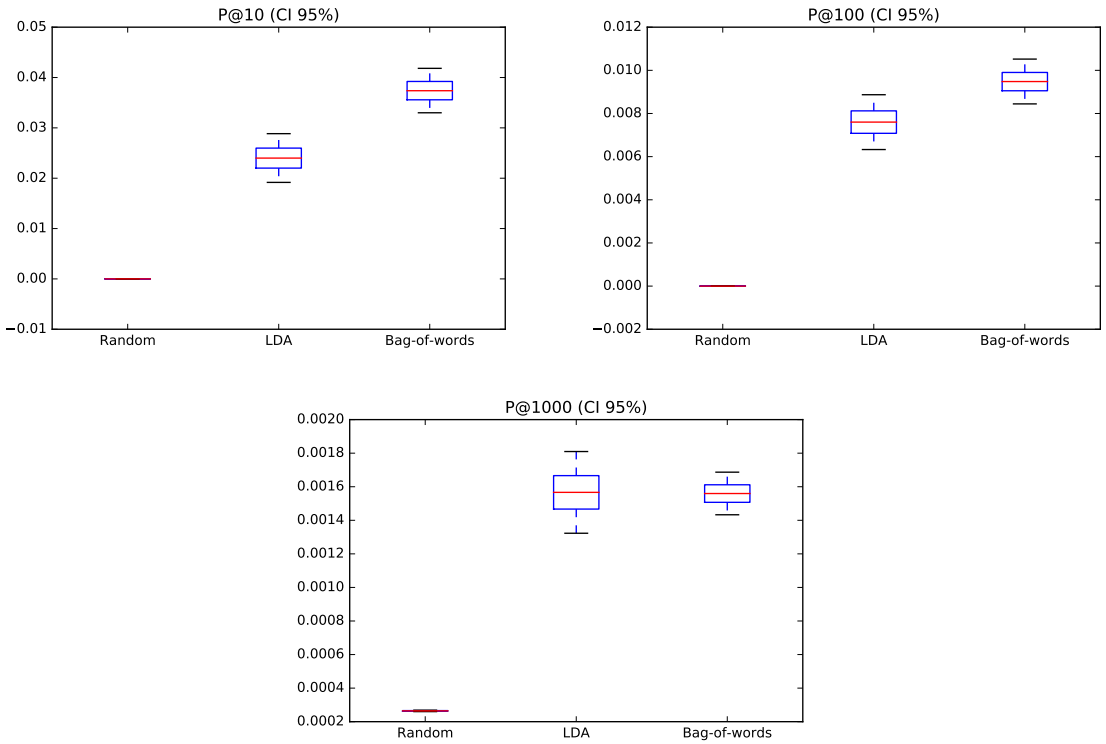


Figure 4: Box plots of final results using P@N as metrics, confidence interval (CI) is 95%

represented in the first topic. The two web pages also had the option to create an account and a sign-in form, represented by the second topic.

4.3.3 Common themes

There were examples of companies that had common themes rather than being competitors.

Example: kncminer.com vs orbitaltraction.com The first company, *kncminer.com*, deals with mining bitcoins and selling hardware for this. The second company, *orbitaltraction.com*, works with rotating machines. On their web pages they explicitly express their environmental concern, which was the reason for the first topic in table 4. The second topic was on the other hand rather generic, consisting largely of words common for most companies' web pages.

4.4 Minimum bisection

The graph to be bisected had approximately 11 000 edges signifying competitor labels. Two algorithms were compared, one greedy algorithm that removed 10-11% of these

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8
weight	iphone	airport	sit	que	delivery	12	skin
id	samsung	booking	ipsum	em	shipping	10	laser
body	galaxy	taxi	lorem	para	free	15	breast
fitness	ipad	book	dolor	um	item	11	surgery
exercise	apple	car	ut	da	items	16	facial
workout	lg	service	amet	com	order	14	cosmetic
loss	tablet	rental	consectetur	uma	service	13	treatment
muscle	phone	travel	elit	mais	customer	17	lift
diet	sony	flight	sed	ou	freight	18	treatments
fat	mini	driver	ex	os	transport	20	plastic

Table 2: Some topics found looking at CrunchBase descriptions concatenated with web content

Topic 1		Topic 2	
english	language	sign	email
translation	español	password	account
spanish	french	login	free
français	chinese	address	terms
german	deutsch	log	privacy
languages	japanese	enter	create
italian	português	use	contact
italiano	portuguese	forgot	register
arabic	russian	send	new

Table 3: Two most common topic and their most common words for *stardoll.com* and *talenthouse.com*

Topic 1		Topic 2	
energy	power	team	clients
technology	waste	experience	business
plant	green	work	technology
carbon	fuel	solutions	development
wind	renewable	help	company
environmental	clean	expertise	approach
production	projects	services	provide
cost	advanced	process	best
sustainable	news	knowledge	strategy
technologies	contact	client	value

Table 4: Two most common topic and their most common words for *kncminer.com* and *orbitaltraction.com*

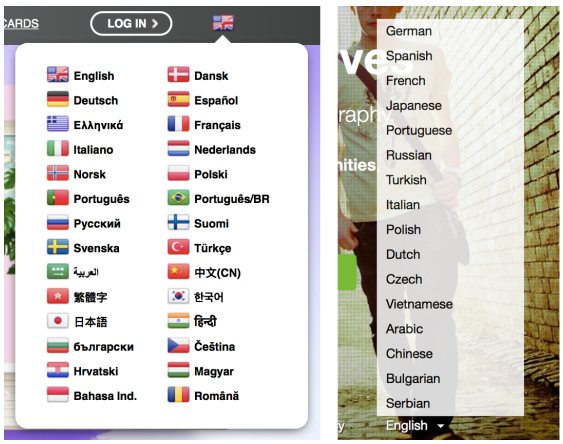


Figure 5: Typical web page content: Choose language

edges, and secondly, a simulated annealing-based algorithm that removed 2-3% of the edges, far outperforming the greedy algorithm.

5 Discussion

5.1 Performance

An interesting discovery was that the cosine similarity, despite its simplicity, overall worked very well. Another unexpected result was that given the test data, LDA-based topic representations performed worse than bag-of-words representations. The benefit of LDA here is that the cosine similarity measure is considerably faster on LDA representations due to lower dimensionality. During model evaluation, using cosine similarity measure was slow enough to calculate on the bag-of-words representations that it became a major problem when producing this report. The compromise between LDA performing worse and bag-of-words being slow is perhaps best decided on a case-by-case basis. An expected result though is that LDA performs significantly better than a totally random algorithm.

5.2 Methodological concerns

This report left a wide array of methods unexplored. In particular no deep learning methods were evaluated, in spite of their recent popularity.

The reason bag-of-words was chosen as a reference model was that since LDA reduces the dimensionality of the bag-of-words representations, it is therefore interesting to investigate how performance is affected after this kind of simplification. Also, two documents about the same topic can use a disjunct set of words but still be described by the same topics using LDA. On the other hand, the bag-of-words representations will end up orthogonal.

Random grid search was performed with a smaller subset of the training data than what was used during final model training, which affects the performance of both models. The optimal *min-df* parameter is likely to change as the number of documents increases. This relationship could be further examined, as it is unclear how this affects each model.

5.3 Other data sources

The main data source used in this report is publicly editable. Competitor labels, business website URLs and descriptions were extracted from this data source, and may therefore suffer from potential mislabeling issues. However, this is mitigated by the fact that the data source is curated by a set of trusted editors. Similarly, only the front page of the business's websites were downloaded. A more thorough approach is possible, and may very well prove beneficial. Some businesses have vague front pages, and only get into detailed descriptions on an "About" page or similar.

5.4 Original aim

When training LDA on the corpus, topics are emerging that can be shared between different documents. A distance measure then measures how topics are co-occuring in two documents. Two companies being competitors are probably closer in topic-space, but the opposite is not necessarily true: adjacent companies are only guaranteed to share some textual themes found by the LDA model, and don't necessarily have to be competitors. When using texts from the web, these common themes can sometimes be non-semantic attributes common to the web such as login forms, language menus, sign up links, terms of use and so on. LDA does its job of finding these co-occuring themes, but doesn't necessarily put competing companies on top in a search result.

5.5 Further work

Although a simplification with lower scores in this report, LDA is indeed interesting in its own right. In particular, it would be interesting to combine with other models to better sort out competitors from a large collection of companies.

没有错误，只有教训
There are no mistakes, only lessons.

– Chinese proverb

References

James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *JMLR*, page 305, 2012.

C.M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, 2006. ISBN 9780387310732. URL `https://books.google.se/books?id=kTNoQgAACAAJ`.

David M. Blei and John D. Lafferty. Topic models. 2009. URL `http://www.cs.princeton.edu/~blei/papers/BleiLafferty2009.pdf`.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003. ISSN 1532-4435. URL `http://www.cs.columbia.edu/~blei/papers/BleiNgJordan2003.pdf`.

Michael R Garey and David S Johnson. Computers and intractability: a guide to the theory of np-completeness. *WH Free. Co., San Fr*, 1979.

G. James, D. Witten, T. Hastie, and R. Tibshirani. *An Introduction to Statistical Learning: with Applications in R*. Springer Texts in Statistics. Springer New York, 2014. ISBN 9781461471370. URL `https://books.google.se/books?id=at1bmAEACAAJ`.

Elaine K.F. Leong, Michael T. Ewing, and Leyland F. Pitt. Analysing competitors' online persuasive themes with text mining. *Marketing Intelligence & Planning*, 22 (2):187–200, 2004. doi: 10.1108/02634500410525850. URL `http://dx.doi.org/10.1108/02634500410525850`.

C.D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. An Introduction to Information Retrieval. Cambridge University Press, 2008. ISBN 9780521865715. URL `http://nlp.stanford.edu/IR-book/`.

Jie Tang, Bo Wang, Yang Yang, Po Hu, Yanting Zhao, Xinyu Yan, Bo Gao, Minlie Huang, Peng Xu, Weichang Li, and Adam K. Usadi. Patentminer: Topic-driven patent analysis and mining. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, pages 1366–1374, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1462-6. doi: 10.1145/2339530.2339741. URL `http://doi.acm.org.focus.lib.kth.se/10.1145/2339530.2339741`.

www.kth.se