Surprisal in Action: A Comparative Study of LDA and LSA for Keyword Extraction

J. Nathanael Philipp
Sächsische Akademie der Wissenschaften
zu Leipzig
Karl-Tauchnitz-Str. 1
04107 Leipzig, Germany
nathanael@philipp.land

Max Kölbl
Osaka University
1-5 Yamadaoka, Suita
565-0871 Osaka, Japan
max.w.koelbl@gmail.com

Michael Richter
Leipzig University
Augustusplatz 10
04109 Leipzig, Germany
mprrichter@gmail.com

Abstract

This study compares two methods of topic detection, *Latent Dirichlet Allocation* (LDA) and *Latent Semantic Analysis* (LSA), by using it in conjuction with the *Topic Context Model* (TCM) on the task of keyword extraction. The surprisal values that TCM outputs based on LDA and LSA are compared, both, directly and as inputs to a Recurrent Neural Network (RNN). While in the direct comparison LSA slightly outperforms LDA, LDA and LSA perform on a par when a Recurrent Neural Network (RNN) is trained with surprisal values. In general: semantic surprisal as input of an RNN improves its performance.

1 Introduction

Keywords serve as classification features of texts and play a crucial role in search engines and natural language understanding (Bharti and Babu, 2017; Karttunen, 1974; Stalnaker, 1977). Following (Çano and Bojar, 2019) and (Tomokiyo and Hurst, 2003), we follow the definition in (Kölbl et al., 2022) and define keywords as nouns or noun phrases, i.e., proper names that can be used as classification features of texts.

This study compares the performance for keyword extraction (KE) of two topic models: *Latent Dirichlet Allocation* (LDA) (Blei et al., 2003) against *Latent Semantic Analysis* (LSA) (Deerwester et al., 1990). These two topic models are applied within the framework of *Information Theory* (Shannon, 1948) and *Surprisal Theory* (Tribus, 1961; Hale, 2001; Jaeger and Levy, 2007) by which operationalisation is carried out using the Topic Context Model (TCM) (Kölbl et al., 2020, 2021; Philipp et al., 2022). TCM has been empirically tested in studies of (Kölbl et al., 2020, 2021) and (Philipp et al., 2022) and allows the calculation of *semantic surprisal*, that is, the degree to which a word is informative in a semantic context.

Following (Kölbl et al., 2021; Philipp et al., 2022), we assume that words with high surprisal values are strong keyword candidates, as they are both unexpected and semantically informative. TCM derives surprisal from topic distributions within a text, necessitating a topic model, LDA or LSA, whose different underlying principles warrant a systematic comparison. The comparison of LDA and LSA is therefore based on the criterion of which of the two topic models the TCM uses to generate the better keywords. Why is it interesting to compare the two topic models? While LDA is probabilistic and assumes a generative process where words are drawn from topics, LSA is deterministic and represents texts as points in a high-dimensional topic space based on word cooccurrences. Both methods have been widely used in information retrieval, yet their respective advantages for KE remain an open question. Our study integrates a Recurrent Neural Network (RNN) with TCM by using *semantic surprisal* as an input feature, allowing us to assess whether this additional information improves keyword prediction while maintaining interpretability. Our research question is: which topic model enables TCM to estimate surprisal more effectively for identifying keywords? While prior studies suggest LDA often outperforms LSA in text mining, the impact of these models on KE within the TCM framework remains unexplored. Since no general benchmark exists for comparing topic models in this regard, we focus on evaluating their KE performance. The structure of the article is as follows: first, we introduce the concept of *surprisal* and outline the theoretical foundation of TCM. This is followed by sections on related work, data resources, and the implementation of LDA and LSA in TCM. Finally, we present our results, followed by a discussion and conclusion.

2 Surprisal and TCM

Surprisal is equivalent to information (Hale, 2001; Levy, 2008): the surprisal of a lingustic unit is estimated from the context of that unit and is thus based on conditional probabilities. In psycholinguistics, the amount of surprisal of a linguisitc unit and the mental effort to process it, are proportional to each other.

Surprisal is interwoven with predictive language processing: given for example a sentence-initial word, a language processor intuitively builds up a prediction what the next word will be (Venhuizen et al., 2019) which is accompanied by a mental *anticipatory pre-activation of information* (Heilbron, 2022). So, if the stimulus word is followed by an expected word, both the surprisal and the processing effort are small, if, in constrast, the stimulus word is followed by an unexpected word, the surprisal is high which requires more processing effort.

The idea of TCM is quite intuitive: the semantic surprisal of a linguistic unit is estimated from the distribution of topics in the environment of that unit. To give a simple but intuitive example, if the word *electric engine* appears in the context of the topics *chocolate* and *caries*, the semantic surprisal of this word is likely to be high and should cause a high processing effort. TCM can consider topics both in an non-local environment of a target word (a corpus, for instance) and topics in local environments (for instance, paragraphs, documents or even sentences).

Why do we utilise surprisal, i.e, information for KE? Our point of departure is Dretske's work (Dretske, 1981) on the flow of information and language comprehension: *Shannon information* is, according to Dretske, the 'raw material' necessary to convey meaning and to build knowledge.

Empirical evidence for a link between Shannon information and meaning comes, in addition to the above mentioned studies of Kölbl et al. (2021) and Philipp et al. (2022), among others from (Melamed, 1997) on semantic entropy, (Aji and Kaimal, 2012; Ravindra et al., 2004) on automatic summarisation, and from (Rubino et al., 2016; Bizzoni and Lapshinova-Koltunski, 2021) on classification of translations.

How can surprisal be defined? The surprisal of a linguistic unit w is proportional to the difficulty or mental effort to process w (Levy, 2008): the smaller the probability of a unit w given a context is, the more surprising and the more informative is w, and the greater the effort is that a language processor needs to process w. This is corroborated through empirical evidence (Brennan et al., 2016; Hale et al., 2015) for lexical surprisal (see amongst others (DeLong et al., 2005; Frank et al., 2015; Smith and Levy, 2008; Szewczyk and Schriefers, 2018; Goodkind and Bicknell, 2018)), for syntactic surprisal (Hagoort et al., 1993; Henderson et al., 2016), and for semantic surprisal (Rabovsky et al., 2018). The surprisal model (Hale, 2001; Levy, 2008) is given in Formula 1:

$$\begin{aligned} \text{difficulty} \propto \text{surprisal} = \\ -\log_2 P(w_i|w_1 \ldots w_{i-1}, \text{CONTEXT}) \end{aligned} \tag{1}$$

In Formula 1, $w_1 \ldots w_{i-1}$ as contexts of w can be n-grams of tokens or n-grams of part-of-speech tags. The former are the basis for the estimation of lexical surprisal, the latter are the basis for the estimation of grammatical surprisal. The variable CONTEXT represents some extra-sentential context, such as sentence structures for syntactic surprisal, or topics, world knowledge, or representation of the discourse for semantic surprisal.

3 Related work

Even though there has been no research comparing LSA and LDA in terms of KE, there are comparative studies from theoretical and empirical points of view. Some of these studies (Griffiths et al., 2007; Niraula et al., 2013) point out the incapability of LSA to grasp polysemy since words are represented as fixed Euclidean points. On the other hand, LDA is based on a generative probabilistic model which follows the Dirichlet multinomial distribution and thus can handle multiple meanings of a word. Based on these distinctive characteristics, (Griffiths et al., 2007) assert that LDA outperforms LSA in predicting associations between words. The most recent experimental research has indeed confirmed the superiority of LDA over LSA, for instance in classifying e-books (Mohammed and Al-augby, 2020; Kalepalli et al., 2020) and on detecting polysemy (Griffiths et al., 2007; Niraula et al., 2013).

Despite these drawbacks of LSA, the empirical study by Bergamaschi and Po (2015) shows that LSA achieves a better performance on movie recommendation than LDA. Papadimitriou et al. (2000) propose three conditions under which LSA, referred to as *Latent Semantic Indexing* (LSI), is capable of capturing the underlying semantics of the corpus, summarised by Anaya (2011, 16-17), i.e., (i) the documents should have an identical or at least very similar writing style, (ii) each document should have a clearly recognisable 'main' topic, and (iii) words should have only a high probability in one topic but a low probability in other topics.

With regard to the theoretical framework of our study, so far, information theory is almost completely disregarded for KE. To the best of our knowledge, only Herrera and Pury (2008) present a model based on information theory. Upon detecting keywords, they assume that highly relevant words should be concentrated in some portions of the text. Their model incorporates the distribution of occurrences of a word in the corpus. However, the calculation of entropy is not based on the semantic model, as in our study by LDA / LSA within the TCM.

With regard to the technique, the application of neural networks to KE achieves successful outcomes, and is state of the art, see for instance (Zhang et al., 2020) who propose the deep neural network model 'target centred LSTM model', which is trained as a binary classifier to determine whether a given word is a keyword or not, and (Grootendorst, 2020) who implemented a BERT-based model capable of KE.

4 Data and Methods

4.1 Text Corpus

As our data resource, we exploit the *Heise*[1] corpus (Philipp et al., 2022). The corpus comprises $100,673$ texts on a variety of topics, often related to technology and telecommunication, with a total of $38,633$ keywords. Each text is prefixed with a set of keywords that we use as a reference set. We assume that thematic diversity will be an advantage for the use of *TCM* since this means a high variety of topics and is expected to optimise the performance of the model. An additional advantage of the *Heise* corpus is that it includes numerous short texts as local environments for the TCM, this is, small environments of target words. In order to avoid memory limits in the case of LDA we use the same subset as Philipp et al. (2022) which consists of randomly drawn samples with a total of $30,284$ texts with $239,065$ words (lemmas, determined by spaCy[2]), and $7,347$ keywords. A randomly chosen sample of 10% of the texts is used as our validation set, the remaining 90% is the training set.

Each text is preceded by a headline, and for $15,454$ of the texts also a short summary, i.e., a *lead* is given, with a average length of one or two sentences.

We treat multiword expressions of named entities as single words and possible keyword candidates. In Philipp et al. (2022), the named-entity recogniser (NER) from *spaCy* classified 122 named entities as specifications of place (LOC), 288 as mixed specifications (MISC), 23 as specifications of organisations (ORG), 40 as specifications of persons (PER), and 6775 were not assigned to any of these categories.

The performance of the NER on the training is given in Philipp et al. (2022, Table 1). The results disclose that the accuracy positively correlates with complexity of the NER models, but all NER models have extremely weak precision and F1 values but better recall. The most complex NER model achieves only a precision of 0.1259. That is to say, only 12.6% of the predicted keywords are real keywords, and the model predicts a large set of keywords whereby the hit rate is relatively low. Recall is 0.3785, which says that almost 38 percent of the keywords are recognised.

4.2 Baselines

NER (Kölbl et al., 2021; Philipp et al., 2022), *TextRank* (Mihalcea and Tarau, 2004) and RNNs without surprisal as input are used as baselines. All named entities that NER identifies, are regarded as keywords. A detailed description of the RNNs can be found in Section 4.4.

TextRank (TR) (Mihalcea and Tarau, 2004) is a modification of Google's *PageRank*-algorithm (Brin and Page, 1998) and performs i.a. summarisation and KE[3]. TR models an entire text as a graph, whereby neighbourhood in the graph and a high weight of the link indicate semantic relatedness.

4.3 Topic Context Model

TCM is an extended topic model: it calculates semantic surprisal based on the output of genuine topic models.

[1] https://heise.de

[2] https://spacy.io

[3] https://github.com/jnphilipp/TextRank

When LDA is employed, TCM calculates surprisal based on probabilities of w in a topic weighted by the probability of that topic in a document. In LSA, only the log-likelihood of w in the main topic in a document is considered. LDA learns the relationship between words and topics in its training phase, while LSA establishes a relationship between documents through the cosine distance.

Figure 1 illustrates TCM, with either LDA or LSA as topic model for the topic distribution.

4.3.1 How LSA works in TCM

LSA reduces the dimension of a text using *Singular Value Decomposition* (SVD). Steinberger et al. (2004) pointed out that SVD "[...] reflects a breakdown of the original document into r linearly-independent base vectors or concepts."

Formula 2 says that, as a first step, the term by sentence-matrix A is decomposed into the three matrices U, Σ and V^T. The columns in U represent the concepts in the Steinberger's quote above (Steinberger et al., 2004):

$$A = U\Sigma V^T \tag{2}$$

The entries of matrix A are the weighted frequencies of terms (rows) in sentences (columns). The number of diagonal elements in Σ is the number of singular vectors representing the semantic dimensions of a text. Following Steinberger et al. (2004), the leftmost column vector in matrix U can be interpreted as the main topic of a document.

Given a finite set M of documents, we can relate the texts semantically by embedding their respective topic vectors into a vector space. (Steinberger et al., 2004) recommend cosine similarity as a measure of similarity of two documents. Since U is unitary, the length of each topic vector is 1, reducing the cosine to a simple scalar product. For a document d with topic vector v_d, we compute the likelihood $P(w_d)$ of a word w_d to appear in d by checking its appearance in other texts, see Formula 3.

$$P(w_d) = \frac{\sum\limits_{s \in M} \delta(w_d, s)\,(1 + \cos\langle v_s, v_d\rangle)}{\sum\limits_{s \in M} 1 + \cos\langle v_s, v_d\rangle} \tag{3}$$

Here, $\delta(w_d, s)$ is 1 if w_d appears in s or 0 if it does not. The corresponding surprisal is then given in Formula 4.

$$surprisal(w_d) = -\log_2 P(w_d|d) \tag{4}$$

We use a scoring function for the RNN, see Formula 5, in order for $surprisal$ values to be between -1 and 1, which we refer to as $\widehat{surprisal}$.

$$\widehat{surprisal}(w_d) = \tanh \frac{surprisal(w_d) - \mu_d}{\sqrt{\sigma_d^2 + \epsilon}} \tag{5}$$

In Formula 5, μ_d is the mean of the $surprisal$ for all the words w in a document d, and σ_d^2 is the variance. To ensure that the standard deviation is not zero we add $\epsilon = 1e^{-7}$.

4.3.2 How LDA works in TCM

LDA is a statistical topic model that tries through a generative process to detect and identify the topics that appear in a document and which words belong to them (Blei et al., 2003). LDA is based on the Dirichlet Distribution with two priors, i.e., *hyperparamaters*: η, the distribution of words in a topic and α, the topic distribution in a document d. The values of these parameters can be chosen deliberately. The only observable phenomena are words w in d, everything else are hidden variables. In d, a word $w_{d,n}$ occurs, given the non-observable assignment of a topic for the n-th word w in d, i.e., $z_{d,n}$, and given the complete set of k topics $\beta_{1:k}$ where β is a distribution of words in a topic k. There is a conditional probability of $z_{d,n}$ given θ_d, i.e., the distribution of topics in document d. The values for β and θ are determined in an iterative process. Let us illustrate this in a fictitious example: The distribution in an (abstract) topic, which is interpreted as 'food', could look as follows (the probability that a word with its meaning is assigned to a topic is given after the comma in each case): topic 'food': chocolate 0.02, butter 0.019, soup 0.021, bicycle 0.00001. 'Bicycle' is thus assigned to the topic 'food' with some much lower probability than 'chocolate', 'butter' and 'soup'.

We define the context as a topic calculated by LDA and calculate the average surprisal $\overline{surprisal}$ for each word, see Formula 6, where n is the number of topics of the LDA. We fixed this at 1000 topics. The calculation is given in Formula 6.

$$\overline{surprisal}(w_d) = -\frac{1}{n}\sum_{i=1}^{n} \log_2 P(w_d|t_i) \tag{6}$$

The term $P(w_d|t_i)$ is the probability of a word w_d given a topic t_i in a document d, which is calculated according to Formula 7. $c_d(w)$ is the frequency of a word w given a document d, $|d|$ is

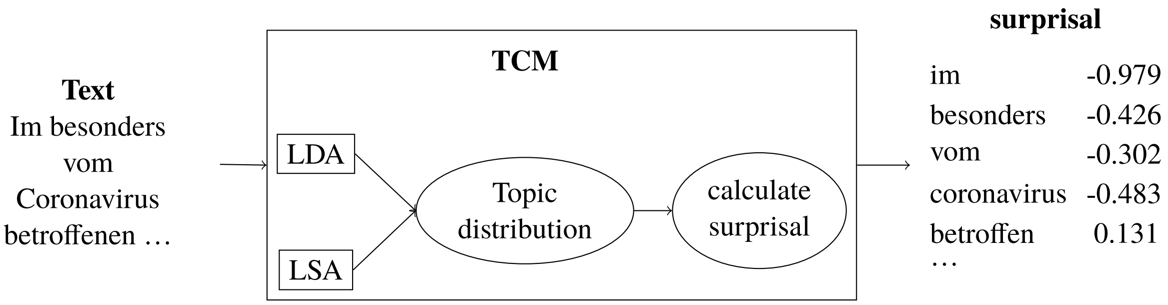

Figure 1: Illustration of the Topic Context Model.

the total number of words in the document d, WT is the normalised word topic distribution of the LDA[4], and $P(t_i|d)$ is probability of a topic t and a document d given by the LDA.

$$P(w_d|t_i) = \frac{c_d(w_d)}{|d|} WT_{w_d,t_i} P(t_i|d) \tag{7}$$

For LDA, we use the same scoring function as for LSA (see Formula 5) except using the $\widehat{surprisal}$ instead of $surprisal$ values.

4.4 Recurrent Neural Network

The RNNs in our study carry out the binary classification whether a word in the input sequence is a keyword or not (see Zhang et al. (2016), Kölbl et al. (2020),Kölbl et al. (2021) and Philipp et al. (2022)).

The architectures of the RNNs in this study is identical to the ones in Philipp et al. (2022). That is to say, for each of the input parts of the texts, namely *headline*, *lead*, and *text*, the RNN has a separate in- and output. In between, one or two bidirectional *Gated Recurrent Units* (GRU) for processing are located. We use identical embeddings and bidirectional GRU(s) for all input types. The architecture of the RNNs is given in Figure 2 that illustrates also the classification-process: when a respective input-word is classified as a keyword, the output is 1, while the output 0 means that the respective word is not classified as a keyword.

Five different input types can be distinguished: (i) text only, (ii) text and information of words $\widehat{surprisal}$ from the topic model LDA, (iii) text and $\widehat{surprisal}$ from the topic model LSA, (iv) – (v) only the $\widehat{surprisal}$ from the TCM model, either from LDA or LSA, respectably. That is to say, the latter two models (iv) – (v) are exclusively trained on the surprisal of words and not on any text. In model (i), each word is represented as a vector with 128 entries from an embedding layer. In the models (ii) – (v), the $\widehat{surprisal}$ values from LDA / LSA in (ii) – (v) are used directly, and in (ii) and (iii) the original embedding vector is concatenated with the $\widehat{surprisal}$ value, resulting in a vector of size 129.

Following in Philipp et al. (2022), for each of the five input types, three configurations with one bidirectional GRU of sizes $(256, 512, 1024)$ and two configurations with two bidirectional GRUs of sizes $(256, 512)$ were trained.

For the input types (i), (ii) and (iv) we report the results from Philipp et al. (2022). We trained, in total, 10 different RNNs ((iii), (v)), each for 15 epochs.

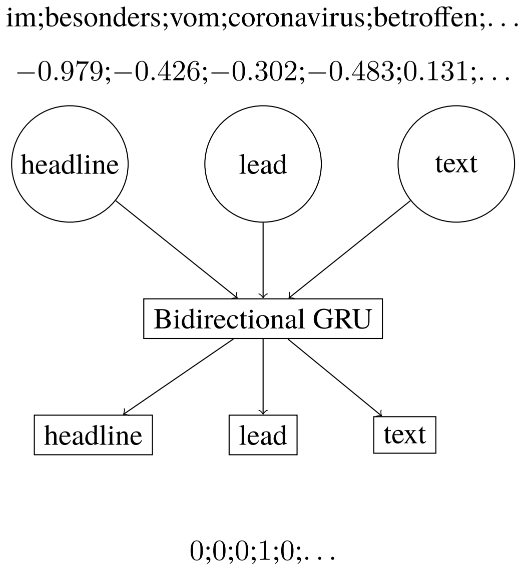

Figure 2: Schematic of the RNN architectures with text and $\widehat{surprisal}$ as input. (Philipp et al., 2022)

[4] `model.components_ / model.components_.sum(axis=1)[:, np.newaxis]` as suggested by https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.LatentDirichletAllocation.html

5 Results

Table 1 gives the results of the baseline models on the validation set from (Philipp et al., 2022), and Table 2 displays the results of the recent study that employs LSA in addition to LDA as part of the TCM.

As in (Philipp et al., 2022), accuracy 1 to 5, precision, recall, and F1-measure are reported, whereby accuracy n means how often at least n keywords are correctly predicted. The F1 score is a measure of the accuracy of a test and is calculated as the harmonic mean of precision and recall. The baselines are NER, TR and RNNs without $surprisal$ as input. For LDA and LSA, the evaluation covers the most informative words of a text, either absolute, e.g., LDA_5 means the top 5 most informative words, or as percentage, e.g., $LDA_{10\%}$ means that the 10% most informative words were selected as keyword candidates. With TR we have the levels TR_5 and TR_{10}, respectively.

Regarding the RNN models, the index refers to, firstly, the input type, e.g., T if it is trained on texts and LDA / LSA means that $\widehat{surprisal}$ from LDA / LSA is part of the input, and secondly, it refers to the layer-sizes of the GRU, i.e., 256, 512, and 1024 respectively.

When no RNN is employed, i.e., with 'pure' TCM, LSA outperforms LDA in direct comparison, in all measures and all rankings. This means that the words with a high $\widehat{surprisal}$ from LSA are more likely keywords than words from the LDA. Further we observe, that neither the top 5 nor the top 10 informative words are better than the 10% most informative words which means that for both LDA and LSA, the most informative words are not necessarily *good* keywords. In general, the 'pure' TCM performs poorly, and is outperformed by other models.

Training the RNNs on both text and $\widehat{surprisal}$ yields the overall best performances in recall, although NER achieves relatively high recall as well. For $a4$ and $a5$ the model RNN$_{T,LDA,512}$ has the overall best results, for recall it is the model RNN$_{T,LSA,1024}$, and for F1 the model RNN$_{LDA,2\times256}$ outperforms the remaining models. However, the differences are small: when training with the $\widehat{surprisal}$ from LDA, the improvement of F1 is about 0.01. The baseline RNN, i.e., trained only on texts, achieves a paper-thin lead in precision of all models - closely followed by the neural networks with $\widehat{surprisal}$ of LDA and LSA.

The model RNN$_{T,LDA,1024}$ performs surprisingly poorly, and we attribute this outcome to a slower learning process compared to the other models. The RNNs trained solely on $\widehat{surprisal}$ from LDA perform in general better than the baseline model 'TR top 5' but not better than 'TR top 10'. This indicates that the $\widehat{surprisal}$ alone performs on the same level as the baselines. In general, we observe that the performance of RNN is improved when the input is supplemented by the input of surprisal values. There is no correlation between performance and the complexity of the network.

Apart from the $a1$ - $a3$ values, where the baselines NER ($a1$) and TextRank ($a2$ and $a3$) achieve the highest values, the RNNs yield the best results in F1 values, which can still be improved by using $\widehat{surprisal}$ as input. NER is the winner in $a1$-accuracy which is probably due to the fact, that most keywords are names. This also explains the *pure* performes for the NER, i.e., the NER is not a very reliable in recognizing the names in the data, which is due to the nature of the data, see (Kölbl et al., 2021; Philipp et al., 2022; Kölbl et al., 2022)

The accuracy values can be interpreted as follows: for about 88% of texts, at least one keyword is a named entity, as far as the spaCy model is concerned. All RNNs trained on $\widehat{surprisal}$ have at some point the same accuracy values, showing that here is their limit, i.e., the RNNs can only learn a certain amount of the pure $\widehat{surprisal}$. TR_{10} achieves high accuracy values $a1 - a3$ and the second highest recall of all models, but only weak precision and a low F1 value. This means that the model fails to detect a satisfactory percentage of keywords from an existing set of keywords. TextRank with the five highest ranked words (TR_5) achieves slightly higher F1 and precision values than when ten hightest ranked words are considered (TR_{10}).

6 Discussion and conclusion

Acceptable results for KE are obtained when $\widehat{surprisal}$ is the input of a recurrent neural network whereby it does almost not matter whether LDA and LSA is employed in the TCM.

So, the first finding of our study is that semantic surprisal improves the performance of RNN, and LDA and LSA are on a par here. In order to recognise the relevance of a word for a text, it is therefore in general beneficial to know the semantic context

Model	a1	a2	a3	a4	a5	Precision	Recall	F1
Baselines								
NER	**0.8771**	0.4528	0.1896	0.0918	0.0558	0.1249	0.3808	0.1784
TR$_5$	0.7781	0.3633	0.1998	0.1724	0.1707	0.2161	0.4341	0.2725
TR$_{10}$	0.8606	**0.5112**	**0.3078**	0.2550	0.2398	0.1371	0.5351	0.2094
RNN$_{T,256}$	0.6902	0.3636	0.2919	0.2847	0.2840	0.5363	0.4669	0.4638
RNN$_{T,512}$	0.6989	0.3795	0.3012	0.2936	0.2926	0.5544	0.4772	0.4773
RNN$_{T,1024}$	0.6929	0.3702	0.2926	0.2853	0.2847	0.5562	0.4697	0.4759
RNN$_{T,2\times256}$	0.6764	0.3557	0.2734	0.2678	0.2672	0.5416	0.4531	0.4622
RNN$_{T,2\times512}$	0.6635	0.3504	0.2751	0.2698	0.2692	**0.5615**	0.4471	0.4660

Table 1: Precision, recall, F1-measure, and the accuracy-values ($a1$ – $a5$) of the baseline models. The best results in for the baselines are underlined and the overall best results in both tables are bold faced. For the RNN models, the subscript first refer to the input and second the model used. T means that the RNN was trained on text, and LDA / LSA refers to what $\widehat{surprisal}$ was part of the input. The numbers refer to the size of the GRU and the number of GRUs. For LDA / LSA / $TextRank$ the subscript refers either to the highest ranked words used, or the percentage of words used.

of this word.

Second, in a direct comparison without a neural network, LSA outperforms LDA. This result is somewhat surprising because, as we mentioned in the introduction, the performance of LDA in practical applications is considered a bit better than that of LSA. We explain this finding that obviously at least one of the three conditions for a good performance of LSA (Papadimitriou et al., 2000; Anaya, 2011) (see 3), i.e., similar writing style ands a clearly recognisable ‘main’ topic in the documents and high probability in one topic but a low probability in other, have been met in our corpus. But it is hard to say which of the three conditions applies.

Besides the better performance of LSA in KE in direct comparison, an additional argument in favour of LSA is that it is ‘nicer’ and more economical in its mathematical computability than LDA. With all due caution, this study provides preliminary evidence that the proposed linear model of topics as vectors in LSA works successfully as part of TCM for KE.

Among the baselines, TR’s performance is very strong in accuracy and recall but the model has shortcomings. The poor performance in precision and the resulting low F1 value can, in our opinion, be explained by TR’s well known weakness that semantic contexts are not considered (Yu et al., 2016).

In general, we observe that the results of our study are mediocre. Why is that the case? The explanation is, as we presume, that semantic surprisal from TCM is derived only from a singular type of context, i.e., topics. Consequently the aim of future research is to build a more complex context model which outputs combined lexical, syntactic and semantic surprisal, and, in addition, situational and knowledge of the world-surprisal (Venhuizen et al., 2019). While the situational and knowledge of the world-surprisal is possibly hard to derive, estimating lexical, syntactic and semantic surprisal should be a realistic goal. We expect that a more complex context model will output better keywords. In addition, it could be tested how LDA and LSA work together in TCM. The conclusions of our study on KE are:

(i) surprisal computed by TCM, improves the performance of RNN, (ii) LSA is slightly superior to LDA, (iii) semantic surprisal alone as features of words provides mediocre results yet, (iv) TCM with both with LDA and LSA is a promising tool for keyword extraction when it enriches the ‘knowledge’ of a neural network.

Limitations

(i) Theoretical: Our work deals only peripherally with mathematical aspects of the comparison between LDA and LSA. We briefly explain the basics of both techniques based on the fundamental distinction that LDA is probabilistic and LSA deterministic.

(ii) Methodological: We have limited ourselves to keyword extraction as an empirical testing procedure of our research question. Other techniques for checking the quality of the ke-

Model	a1	a2	a3	a4	a5	Precision	Recall	F1
Topic Context Models with LDA and LSA								
LDA$_5$	0.0281	0.0043	0.0036	0.0033	0.0033	0.0059	0.0126	0.0076
LDA$_{10}$	0.0888	0.0185	0.0135	0.0135	0.0132	0.0096	0.0427	0.0152
LDA$_{10\%}$	0.1219	0.0228	0.0162	0.0162	0.0159	0.0107	0.0552	0.0173
LDA$_{20\%}$	0.2665	0.0816	0.0532	0.0479	0.0476	0.0128	0.1371	0.0228
LDA$_{30\%}$	0.3653	0.1374	0.0869	0.0766	0.0753	0.0124	0.2018	0.0230
LSA$_5$	0.0535	0.0149	0.0132	0.0129	0.0129	0.0113	0.0290	0.0154
LSA$_{10}$	0.1433	0.0469	0.0370	0.0363	0.0363	0.0160	0.0799	0.0256
LSA$_{10\%}$	0.1780	0.0528	0.0376	0.0367	0.0363	0.0171	0.0948	0.0279
LSA$_{20\%}$	0.4326	0.2057	0.1526	0.1410	0.1404	0.0228	0.2725	0.0411
LSA$_{30\%}$	$\underline{0.6380}$	$\underline{0.3785}$	$\underline{0.2949}$	$\underline{0.2711}$	$\underline{0.2682}$	$\underline{0.0246}$	$\underline{0.4465}$	$\underline{0.0459}$
Recurrent Neural Networks with LDA								
RNN$_{LDA,256}$	0.3362	0.1159	0.1143	0.1143	0.1143	0.3249	0.2038	0.2355
RNN$_{T,LDA,256}$	$\underline{0.7028}$	0.3768	0.3015	0.2933	0.2929	0.5571	0.4785	0.4798
RNN$_{LDA,512}$	0.2645	0.0908	0.0888	0.0888	0.0888	0.2518	0.1605	0.1845
RNN$_{T,LDA,512}$	$\underline{0.7028}$	0.3768	$\underline{0.3048}$	**0.2959**	**0.2952**	0.5576	$\underline{0.4792}$	0.4812
RNN$_{LDA,1024}$	0.3445	0.1159	0.1133	0.1133	0.1133	0.3245	0.2069	0.2371
RNN$_{T,LDA,1024}$	0.6767	0.3639	0.2860	0.2807	0.2801	0.5423	0.4602	0.4645
RNN$_{LDA,2\times256}$	0.3554	0.1219	0.1189	0.1189	0.1189	0.3349	0.2144	0.2455
RNN$_{T,LDA,2\times256}$	0.6988	$\underline{0.3791}$	0.3025	0.2932	0.2926	$\underline{0.5586}$	0.4775	**0.4817**
RNN$_{LDA,2\times512}$	0.3326	0.1169	0.1139	0.1139	0.1139	0.2299	0.3094	0.2031
RNN$_{T,LDA,2\times512}$	0.6929	0.3686	0.2933	0.2853	0.2847	0.5527	0.4696	0.4751
Recurrent Neural Networks with LSA								
RNN$_{LSA,256}$	0.2503	0.0803	0.0803	0.0803	0.0803	0.25	0.1491	0.1767
RNN$_{T,LSA,256}$	0.6823	0.3613	0.2870	0.2810	0.2804	0.5461	0.4620	0.4680
RNN$_{LSA,512}$	0.2345	0.0836	0.0822	0.0822	0.0822	0.2282	0.1446	0.1669
RNN$_{T,LSA,512}$	0.6879	$\underline{0.3771}$	$\underline{0.2966}$	$\underline{0.2906}$	$\underline{0.2900}$	0.5510	0.4708	$\underline{0.4745}$
RNN$_{LSA,1024}$	0.4128	0.1404	0.1328	0.1328	0.1328	0.2774	0.3722	0.2472
RNN$_{T,LSA,1024}$	0.6952	0.3748	0.2962	0.2900	0.2890	0.4739	**0.5483**	0.4731
RNN$_{LSA,2\times256}$	0.2490	0.0925	0.0908	0.0908	0.0908	0.2389	0.1552	0.1768
RNN$_{T,LSA,2\times256}$	0.6863	0.3616	0.2906	0.2837	0.2834	$\underline{0.5591}$	0.4638	0.4721
RNN$_{LSA,2\times512}$	0.1156	0.0423	0.0423	0.0423	0.0423	0.1151	0.0723	0.0842
RNN$_{T,LSA,2\times512}$	$\underline{0.7034}$	0.3699	0.2949	0.2867	0.2863	0.5313	0.4756	0.4671

Table 2: Precision, recall, F1-measure, and the accuracy-values ($a1 - a5$) of TCMs and RNNs. The best results in the respective blocks are underlined and the overall best esults in both tables are bold faced. For the RNN models, the subscript first refer to the input and second the model used. T means that the RNN was trained on text, and LDA / LSA refers to what $\widehat{surprisal}$ was part of the input. The numbers refer to the size of the GRU and the number of GRUs. For LDA / LSA / $TextRank$ the supscript refers either to the highest ranked words used, or the percentage of words used.

words are conceivable such as experiments with raters or experiments on language generation based on the keywords by LLM, as are other techniques for answering our research question from text mining and information extraction such as summarisation. These are tasks of future work and were beyond the scope of this work.

(iii) Empirical: Our study is based on a single corpus that tends to contain texts on the topic of ‘computer technology’. In future, corpora with mixed thematic orientations would be desirable in order to avoid bias.

References

Subhanpurno Aji and Ramachandra Kaimal. 2012. Document summarization using positive pointwise mutual information. *International Journal of Computer Science & Information Technology*, 4(2):47.

Leticia H. Anaya. 2011. *Comparing Latent Dirichlet Allocation and Latent Semantic Analysis as Classifiers*. ProQuest LLC. Publication Title: ProQuest LLC.

Sonia Bergamaschi and Laura Po. 2015. Comparing LDA and LSA Topic Models for Content-Based Movie Recommendation Systems. In *Web Information Systems and Technologies*, Lecture Notes in Business Information Processing, pages 247–263, Cham. Springer International Publishing.

Santosh Kumar Bharti and Korra Sathya Babu. 2017. Automatic keyword extraction for text summarization: A survey. *arXiv preprint arXiv:1704.03242*.

Yuri Bizzoni and Ekaterina Lapshinova-Koltunski. 2021. Measuring translationese across levels of expertise: Are professionals more surprising than students? In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 53–63, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Jonathan R Brennan, Edward P Stabler, Sarah E Van Wagenen, Wen-Ming Luh, and John T Hale. 2016. Abstract linguistic structure correlates with temporal activity during naturalistic comprehension. *Brain and language*, 157:81–94.

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117.

Erion Çano and Ondřej Bojar. 2019. Keyphrase generation: A multi-aspect survey. In *2019 25th Conference of Open Innovations Association (FRUCT)*, pages 85–94. IEEE.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407.

Katherine A DeLong, Thomas P Urbach, and Marta Kutas. 2005. Probabilistic word pre-activation during language comprehension inferred from electrical brain activity. *Nature neuroscience*, 8(8):1117–1121.

Fred Dretske. 1981. *Knowledge and the Flow of Information*. MIT Press.

Stefan L Frank, Leun J Otten, Giulia Galli, and Gabriella Vigliocco. 2015. The erp response to the amount of information conveyed by words in sentences. *Brain and language*, 140:1–11.

Adam Goodkind and Klinton Bicknell. 2018. Predictive power of word surprisal for reading times is a linear function of language model quality. In *Proceedings of the 8th workshop on cognitive modeling and computational linguistics (CMCL 2018)*, pages 10–18.

Thomas L. Griffiths, Mark Steyvers, and Joshua B. Tenenbaum. 2007. Topics in semantic representation. *Psychological Review*, 114(2):211–244.

Maarten Grootendorst. 2020. Keybert: Minimal keyword extraction with bert.

Peter Hagoort, Colin Brown, and Jolanda Groothusen. 1993. The syntactic positive shift (sps) as an erp measure of syntactic processing. *Language and cognitive processes*, 8(4):439–483.

John Hale. 2001. A probabilistic earley parser as a psycholinguistic model. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, pages 1–8. Association for Computational Linguistics.

John Hale, David Lutz, Wen-Ming Luh, and Jonathan Brennan. 2015. Modeling fmri time courses with linguistic structure at various grain sizes. In *Proceedings of the 6th workshop on cognitive modeling and computational linguistics*, pages 89–97.

Micha Heilbron. 2022. *Getting ahead: Prediction as a window into language, and language as a window into the predictive brain*. Ph.D. thesis, [Sl]:[Sn].

John M Henderson, Wonil Choi, Matthew W Lowder, and Fernanda Ferreira. 2016. Language structure in the brain: A fixation-related fmri study of syntactic surprisal in reading. *Neuroimage*, 132:293–300.

Juan P. Herrera and Pedro A. Pury. 2008. Statistical keyword detection in literary corpora. *The European Physical Journal B*, 63(1):135–146. ArXiv: cs/0701028.

T Florian Jaeger and Roger P Levy. 2007. Speakers optimize information density through syntactic reduction. In *Advances in neural information processing systems*, pages 849–856.

Yaswanth Kalepalli, Shaik Tasneem, Pasupuleti Durga Phani Teja, and Suneetha Manne. 2020. Effective Comparison of LDA with LSA for Topic Modelling. In *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, pages 1245–1250.

Lauri Karttunen. 1974. Presupposition and linguistic context. *Theoretical linguistics*, 1(1-3):181–194.

Max Kölbl, Yuki Kyogoku, J. Nathanael Philipp, Michael Richter, Clemens Rietdorf, and Tariq Yousef. 2020. Keyword Extraction in German: Information-theory vs. Deep Learning. In *Proceedings of the 12th International Conference on Agents and Artificial Intelligence - Volume 1: NLPinAI*, pages 459–464. INSTICC, SciTePress.

Max Kölbl, Yuki Kyogoku, J. Nathanael Philipp, Michael Richter, Clemens Rietdorf, and Tariq Yousef. 2021. The semantic level of shannon information: Are highly informative words good keywords? a study on german. In Roussanka Loukanova, editor, *Natural Language Processing in Artificial Intelligence - NLPinAI 2020*, volume 939 of *Studies in Computational Intelligence (SCI)*, pages 139–161. Springer International Publishing.

Max Kölbl, Yuki Kyogoku, J. Nathanael Philipp, Michael Richter, Clements Rietdorf, and Tariq Yousef. 2022. Beyond the failure of direct-matching in keyword evaluation: A sketch of a graph based solution. *Frontiers in Artificial Intelligence*, 5.

Roger Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177.

I Dan Melamed. 1997. Measuring semantic entropy. In *Tagging Text with Lexical Semantics: Why, What, and How?*

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.

Shaymaa H. Mohammed and Salam Al-augby. 2020. LSA & LDA topic modeling classification: comparison study on e-books. *Indonesian Journal of Electrical Engineering and Computer Science*, 19(1):353–362. Number: 1.

Nobal Niraula, Rajendra Banjade, Dan Ştefănescu, and Vasile Rus. 2013. Experiments with Semantic Similarity Measures Based on LDA and LSA. In *Statistical Language and Speech Processing*, Lecture Notes in Computer Science, pages 188–199, Berlin, Heidelberg. Springer.

Christos H. Papadimitriou, Prabhakar Raghavan, Hisao Tamaki, and Santosh Vempala. 2000. Latent Semantic Indexing: A Probabilistic Analysis. *Journal of Computer and System Sciences*, 61(2):217–235.

J. Nathanael Philipp, Max Kölbl, Yuki Kyogoku, Tariq Yousef, and Michael Richter. 2022. One step beyond: Keyword extraction in german utilising surprisal from topic contexts. In *Intelligent Computing*, pages 774–786, Cham. Springer International Publishing.

Milena Rabovsky, Steven S Hansen, and James L McClelland. 2018. Modelling the n400 brain potential as change in a probabilistic representation of meaning. *Nature Human Behaviour*, 2(9):693–705.

G Ravindra, N Balakrishnan, and KR Ramakrishnan. 2004. Multi-document automatic text summarization using entropy estimates. In *International Conference on Current Trends in Theory and Practice of Computer Science*, pages 289–300. Springer.

Raphael Rubino, Ekaterina Lapshinova-Koltunski, and Josef van Genabith. 2016. Information density and quality estimation features as translationese indicators for human translation classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 960–970, San Diego, California. Association for Computational Linguistics.

Claude Elwood Shannon. 1948. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423.

Nathaniel J Smith and Roger Levy. 2008. Optimal processing times in reading: a formal model and empirical investigation. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 30.

Robert Stalnaker. 1977. Pragmatic presuppositions. In *Proceedings of the Texas conference on per~ formatives, presuppositions, and implicatures. Arlington, VA: Center for Applied Linguistics*, pages 135–148. ERIC.

Josef Steinberger, Karel Jezek, and 1 others. 2004. Using latent semantic analysis in text summarization and summary evaluation. *Proc. ISIM*, 4:93–100.

Jakub M Szewczyk and Herbert Schriefers. 2018. The n400 as an index of lexical preactivation and its implications for prediction in language comprehension. *Language, Cognition and Neuroscience*, 33(6):665–686.

Takashi Tomokiyo and Matthew Hurst. 2003. A language model approach to keyphrase extraction. In *Proceedings of the ACL 2003 workshop on Multiword expressions: analysis, acquisition and treatment*, pages 33–40.

Myron Tribus. 1961. Information theory as the basis for thermostatics and thermodynamics. *Journal of Applied Mechanics*, 28(1):1–8.

Noortje J Venhuizen, Matthew W Crocker, and Harm Brouwer. 2019. Expectation-based comprehension: Modeling the interaction of world knowledge and linguistic experience. *Discourse Processes*, 56(3):229–255.

Shan-shan Yu, Jin-dian Su, and Peng-fei Li. 2016. Improved textrank-based method for automatic summarization. *Computer Science*, 43(6):240–247.

Qi Zhang, Yang Wang, Yeyun Gong, and Xuan-Jing Huang. 2016. Keyphrase extraction using deep recurrent neural networks on Twitter. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 836–845.

Yu Zhang, Mingxiang Tuo, Qingyu Yin, Le Qi, Xuxiang Wang, and Ting Liu. 2020. Keywords extraction with deep neural network model. *Neurocomputing*, 383:113–121.