To Extend or Not to Extend? Context-specific Corpus Enrichment

Felix Kuhr, Tanya Braun, Magnus Bender, and Ralf Möller

Institute of Information Systems, University of Lübeck, Lübeck
{kuhr, braun, bender, moeller}@ifis.uni-luebeck.de

Abstract. An agent in pursuit of a task may work with a corpus of documents with linked subjective content descriptions. Faced with a new document, an agent has to decide whether to include that document in its corpus or not. Basing the decision on only words, topics, or entities, has shown to not lead to a balanced performance for varying documents. Therefore, this paper presents an approach for an agent to decide if a new document adds value to its existing corpus by combining texts and content descriptions. Furthermore, an agent can use the approach as a starting point for high quality content descriptions for new documents. A case study shows the effectiveness of our approach given varying types of new documents.

Keywords: Subjective content description · Text mining

1 Introduction

An agent that pursues a defined task or goal may work with a set of documents (corpus) as a form of reference library. A person assembling a range of scientific articles as related work describes such a setting, with the person as the agent, the compiling of related articles as the task, and the articles as the library. From an agent-theoretic perspective, an agent is a rational, autonomous unit acting in a world, perceived through sensors, fulfilling a task, e.g., an agent providing document retrieval services given specific requests from users. For more effective performance, the documents may be annotated with subjective content descriptions (SCDs), which the agents expects to be relevant to its task. The task provides a context in which SCDs add value for the agent. E.g., notes added to specific sentences of an article may provide explanations or references. Thus, SCDs add information relevant for the task or goal and that information has a connection to specific words in the document.

But, what should an agent do if presented with a new document, which typically has no SCDs? Without having thoroughly processed the new document, the question for the agent is: Does that document have anything of value to add in the given context? The problem is a decision making problem: Should the agent extend its reference library with the new document or should it not?

We aim at providing an approach to making a multi-dimensional decision in the context of the task. We focus on the SCDs in the corpus, which are specific to the task and have links to words in the documents. We model that SCDs generate the words of a document. The links between an SCD and specific words are modelled by a sequence of words in a window, with an SCD located at the center of the window. Words closer to the SCD location are more likely generated by the SCD than words farther away. The key idea for determining if a document adds value to the corpus is to measure how much of a new document without any SCDs can one generate with the existing SCDs in the corpus.

We build an SCD-word distribution for each SCD describing how likely an SCD generates each word of the vocabulary of the corpus. The problem turns into finding most probable SCDs (MPSCDs) for a new document given the SCD-word distributions of the existing SCDs. Given MPSCDs of a new document and their probabilities, an agent decides whether to extend the reference library with that document or not. Solving the problem of finding MPSCDs exactly is infeasible as the vocabulary of a corpus is huge with a large number of annotations. Therefore, we work with the similarity between vectors of SCD-word distributions and vector representations of estimated windows of the new document. We use the cosine similarity because the vectors are sparse and cosine similarity has a low complexity for sparse vectors. An agent identifies those SCDs that have vector representations most similar to the window vectors, assuming that those SCDs are the most probable ones based on the statistics of the corpus. Using the similarities of the chosen SCDs, an agent then computes indicators such as minimum, maximum, and average similarity. Based on that indicators, it decides whether the new document is too similar or too dissimilar compared to the documents in its corpus to contribute anything useful in the context of its task. If the agent decides to extend its library with the new document, it can even choose to retain SCDs with highest similarity, possibly adapting them, and then use them as a basis for further enrichment, automatic [15,8] or manual.

Specifically, the contributions of this paper are: (i) solving the problem of finding MPSCD for a new document by estimating them based on SCD-vector comparisons, (ii) providing a decision making procedure based on MPSCD, and (iii) a case study regarding decision making based on MPSCD given varying new documents. Additionally, we look at two special cases: (i) For large corpora containing documents from various topics, filtering a corpus based on topics decreases the number of documents to consider. (ii) For adapting estimated MPSCDs, an expectation-maximisation approach allows for optimizing window size and positions, e.g., if interested in retaining SCDs for new documents.

The remainder of this paper starts with a look at related work. Then, we specify notations for documents and SCDs. Next, we present our contributions, followed by a case study. The paper ends with a conclusion and future work.

2 Related Work

Over the past 20 years, a considerable number of automatic (semantic) annotation systems have been developed. The systems extract named entities from text of documents and add so-called *semantic annotations* from externally available common-sense knowledge bases, enriching the documents with machine-processable data. Some well-known automatic annotation systems are YEDDA [16], MINTE [4], OpenCalais [12], YAGO[14], KDTA [11], or GATE [5]. Some well-known sources of common-sense knowledge are DBpedia [9], NELL [3], and KnowledgeVault [6]. Named entities represent the link between documents and common-sense knowledge and link prediction is used to identify semantic annotations for an entity. Generally, link prediction describes the task of estimating the likelihood of a link (relation) existing between nodes (entities), given the links and attributes of nodes within a graph [7]. These annotation systems aim at developing a knowledge graph augmented with data from external sources. The systems efficiently solve their underlying problem. However, we investigate a different problem, deciding if a new document provides value to an agent.

Surveying methods of text mining, one can base a decision on different aspects, e.g., (i) similarity of text in the spirit of tf.idf [13], comparing a vector representation of a new document with vector representations of the documents in the corpus, (ii) similarity of topics in the spirit of latent Dirichlet allocation (LDA) [2], comparing an estimated topic distribution of a new document with topic distributions of the documents in the corpus, or (iii) entity matching [10] using named-entity recognition (NER), comparing entities (and relations) retrieved from the new document with entities (and relations) of the SCDs in the corpus. All three approaches carry drawbacks: The first two, based on bag-of-words, ignore SCDs and the order of words. Additionally, they make it hard to model that a document has to add value, i.e., not be a rephrased copy of an existing document or contain only unrelated data. The last approach has the problem that NER tools might not output annotations in the context of the task, which may lead to very few matches with SCDs of the corpus. Additionally, the decision in each case is a one-dimensional decision, based on one feature of the documents. Therefore, we aim at providing an approach to make a multi-dimensional decision that considers the context of the task.

3 Preliminaries

This section specifies notations of the technical framework for this paper and gives a brief overview of LDA, which is used for filtering large corpora (see Section 4.4) and is part of our case study in Section 5.

3.1 Notation

We define the following terms to formalize the setting of a corpus of documents, each document associated with a repository of additional data, i.e., SCDs.

– A word w is a basic unit of discrete data from a vocabulary $\mathcal{V} = (w_1, \ldots, w_V)$. Each w is represented as a unit-basis vector of length V that has a value of 1 where $w = w_v$ and 0's otherwise.
– A document d is a sequence of words $(w_1, \ldots, w_D)$ where each w_d is from $\mathcal{V}$. The expression $words(d)$ refers to the number of words in d.

- A corpus $\mathcal{D}$ is a set of N documents $\{d_1, \dots, d_N\}$.
- Each document $d \in \mathcal{D}$ has a link to a document-specific repository g containing a set of SCDs $\{(t_j, \{\rho_i\}_{i=1}^{l})\}_{j=1}^{s}$. SCDs can take any form. As such, their formats may be highly diverse. A standardized format would be the Resource Description Framework (RDF) but, for our main contributions, the specific format is irrelevant. Each t is associated with a set of positions $\{\rho_i\}_{i=1}^{l}$ in d where ρ_i refers to the ρ_i'th word in d. Given a document d or repository g, the terms $g(d)$ and $d(g)$ refer to the linked document and repository, respectively. The set of all SCDs of documents in $\mathcal{D}$ is given by $g(\mathcal{D}) = \bigcup_{d \in \mathcal{D}} g(d)$.
- An SCD window $win_{d,t,\rho}$ refers to the words in d that surround the position ρ of $t \in g(d)$, i.e., $win_{d,t,\rho} = (w_{(\rho-i)}, ..., w_{\rho}, ..., w_{(\rho+i)})$, $i \in \mathbb{N}$ if ρ marks the middle of the window. The position of a word $w \in win_{d,t,\rho}$ is given by $pos(w, win_{d,t,\rho})$ (0-based numbering). The size of $win_{d,t,\rho}$ is given by $s(win_{d,t,\rho})$, i.e., $s(win_{d,t,\rho}) = 2i + 1$ if ρ marks the middle of the window.
- Each word $w \in win_{d,t,\rho}$ is linked to an influence value $I(w, win_{d,t,\rho})$. The closer a word w is positioned to the position ρ of t, the higher is the influence value $I(w, win_{d,t,\rho})$. The influence value $I(w, win_{d,t,\rho})$ of w is given by the probability of the Binomial distribution at position $pos(w, win_{d',t,\rho})$, i.e.,

$$I(w, win_{d,t,\rho}) = \binom{n}{k} \cdot \pi^k \cdot (1 - \pi)^{n-k}, \quad (1)$$

  where $n = s(win_{d',t,\rho}) - 1$, $k = pos(w, win_{d',t,\rho})$, and $\pi = \frac{\rho}{n}$, i.e., $\pi = 0.5$ if t is at the center of $win_{d,t,\rho}$ and influence values to the left and right of ρ should be symmetric. The binomial distribution yields a probability for each word $w \in win_{d,t,\rho}$ that is higher the closer w is to the position of t.

3.2 Latent Dirichlet Allocation

LDA [2], a well-known statistical technique, assumes that documents in a corpus $\mathcal{D}$ represent a mixture of topics where each topic is characterized by a distribution of words from a vocabulary $\mathcal{V}$ of $\mathcal{D}$. LDA generates a topic model from the documents in $\mathcal{D}$, learning latent structures of two forms, (i) a *document-topic distribution* θ for each document $d \in \mathcal{D}$, i.e., the degree to which the content of d relates to each topic in a set of topics and (ii) a *topic-word distribution* ϕ describing the probability of each word from $\mathcal{V}$ occurring in each topic. Both the document-topic distribution and the word-topic distribution depend on the documents in $\mathcal{D}$.

Next, we present the main contributions of this paper, context-specific corpus enrichment based on MPSCDs.

4 Context-specific Corpus Enrichment

This section provides the theoretical foundations for MPSCDs and presents our approach to estimating MPSCDs. Additionally, it looks into two special cases. This section ends with decision making using MPSCDs.

4.1 Foundations of SCDs

This paragraph formalizes the link between SCDs and the words of a document as a basis for MPSCDs. Specifically, we model that SCDs generate the words of a document. Each document $d \in \mathcal{D}$ has a link to its repository $g(d)$, containing SCDs that themselves are linked to positions in d. Words closer to an SCD location are more likely generated by the SCD than words farther away.

Mathematically, we represent each SCD as a vector of length n, where $n = |\mathcal{V}(\mathcal{D})|$ and each vector entry refers to a word in $\mathcal{V}(\mathcal{D})$. The entry itself is a probability that describes how likely it is that the corresponding SCD generates the word, yielding an SCD-word distribution for each SCD. We represent SCD-word distributions for all m SCDs in $g(\mathcal{D})$ by an $m \times n$ matrix $\delta(\mathcal{D})$, with the SCD-word distribution vectors forming the rows of the matrix:

$$\delta(\mathcal{D}) = \begin{array}{c|ccccc|} & w_1 & w_2 & w_3 & \cdots & w_n \\ \hline t_1 & v_{1,1} & v_{1,2} & v_{1,3} & \cdots & v_{1,n} \\ t_2 & v_{2,1} & v_{2,2} & v_{2,3} & \cdots & v_{2,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ t_m & v_{m,1} & v_{m,2} & v_{m,3} & \cdots & v_{m,n} \end{array}$$

We can fill $\delta(\mathcal{D})$ based on the documents in $\mathcal{D}$ and their linked SCDs. Using a maximum-likelihood strategy, one counts for each SCD t the number of occurrences of each word w in the windows $win_{d,t,\rho}$ of t over all documents and all positions. The occurrences are weighted by their influence value $I(w, win_{d,t,\rho})$.

Algorithm 1 shows a description of forming matrix $\delta(\mathcal{D})$, in which $\delta(\mathcal{D})[t][w]$ refers to entry at the intersection of the row of t and the column of w. The term $\delta(\mathcal{D})[t]$ refers to the complete row of t. The outer loop goes through each SCD with the three inner loops counting the weighted occurrences of words in windows of each document. At the end of each outer loop iteration, the SCD-word distribution of the current t is normalized to yield a probability distribution for each SCD over the complete vocabulary. Formally, normalization is given by

$$v_{i,j} = \frac{v_{i,j}}{\sum_{k=1}^{n} v_{i,k}}. \tag{2}$$

To illustrate line 7 of Alg. 2, consider the following example. Assume that in document d_1, there is a window $win_{d_1,t_1,\rho}$ for SCD t_1

$$win_{d_1,t_1,\rho} = (w_{21}, w_4, w_8, \underline{w_{15}}, w_{16}, w_{23}, w_{42}) \tag{3}$$

$$(0.015625, 0.09375, 0.234375, \underline{0.3125}, 0.234375, 0.09375, 0.015625) \tag{4}$$

with t_1 positioned at the center (w_{15}, underlined) and the influence values in Eq. (4) based on Eq. (1) with $n = s(win_{d_1,t_1,\rho}) - 1 = 6$, $k \in \{0, \ldots, 6\}$, and entry positions corresponding to positions in Eq. (3). Based on the innermost loop of Alg. 1, seven entries of $\delta(\mathcal{D})$ are updated, e.g., for w_{21} at position 0:

$$\delta(\mathcal{D})[t_1][w_{21}] \mathrel{+}= 0.015625$$

Algorithm 1 Forming SCD-word distribution matrix $\delta(\mathcal{D})$

```
1: function buildMatrix(Corpus $\mathcal{D}$)
2:     Initialize an $m \times n$ matrix $\delta(\mathcal{D})$ with zeros
3:     for each $t \in g(\mathcal{D})$ do
4:         for each $d \in \mathcal{D}$ do
5:             for each $win_{d,t,\rho} \in d$ do                ▷ Iterates over $\rho$
6:                 for each $w \in win_{d,t,\rho}$ do
7:                     $\delta(\mathcal{D})[t][w]$ += $I(w, win_{d,t,\rho})$    ▷ For $I$, see Eq. (1)
8:         Normalize $\delta(\mathcal{D})[t]$                      ▷ See Eq. (2)
9:     return $\delta(\mathcal{D})$
```

where $\delta(\mathcal{D})[t_1][w_{21}]$ refers to $v_{1,21}$, which is incremented by 0.015625. Algorithm 1 updates $\delta(\mathcal{D})$ for the remaining words and then continues with the next window. When Alg. 1 is finished with d_1, it moves on to the next document, going through the windows of the next document. After iterating over all documents, Alg. 1 repeats going through all documents and their windows for the remaining SCDs.

The model behind a SCD-word distribution matrix $\delta(\mathcal{D})$ is generative as one could now choose $M \ll m$ SCDs and sample a new document based on the chosen SCDs. Given the generative nature, we are now interested in the most likely SCDs to have generated words of a new document.

Thus, we look at the problem of finding those SCDs that are most probable given the words of a new document.

4.2 The MPSCD Problem

Generally, the MPSCD problem asks for the M most probable SCDs for a document d' given the SCD-word distribution matrix $\delta(\mathcal{D})$ and the words in d':

$$\underset{t_1,\ldots,t_M \in g(\mathcal{D})}{\arg\max} \; P(t_1, \ldots, t_M | d', \delta(\mathcal{D})) \tag{5}$$

As we do not model an influence of one SCD on the next and as we place the M windows evenly distributed over d', we can simplify Eq. (5) as follows

$$\underset{t_1 \in g(\mathcal{D})}{\arg\max} P(t_1 | win_{d',t_1,\rho}, \delta(\mathcal{D}) \cup \cdots \cup \underset{t_M \in g(\mathcal{D})}{\arg\max} P(t_M | win_{d',t_M,\rho}, \delta(\mathcal{D})) \tag{6}$$

That is there are M windows $win_{t,d',\rho}$ and for each window individual MPSCDs are estimated. The intuition is as follows: If d' is a document from $\mathcal{D}$ or a close variation of the documents in $\mathcal{D}$, then Eq. (5) yields MPSCDs with high probabilities. If d' is an unknown document, the resulting MPSCDs vary in their probability. If the vocabulary or word composition is very different, the probabilities are very low on average. The closer the vocabulary and word composition of d' get to the characteristics of $\mathcal{D}$, the higher the probabilities are. Determining if a document adds value to a corpus is based on of how much of a new document can one generate with high probability given $\delta(\mathcal{D})$ and what is too much.

Unfortunately, solving the MPSCD problem is intractable as n and m are typically very large, which is why we estimate probability with similarity.

4.3 Estimating MPSCDs

Based on the statistics of the corpus, we estimate Eq. (6) for an SCD t in a window $win_{t,d',\rho}$ by determining the SCD in $\delta(\mathcal{D})$ with the most similar distribution compared to a vector representation of $win_{t,d',\rho}$ using influence values.

The setting is as follows: Given a new document d' and the SCD-word distribution matrix $\delta(\mathcal{D})$, we estimate M MPSCDs. Based on M, M windows $win_{t,d',\rho}$ lie over the text of d' with a window size of $\sigma = \frac{words(d')}{M}$ and positions ρ starting at $\frac{\sigma}{2}$ and incrementing by σ. For each $win_{t,d',\rho}$, the SCD is unknown at the start, i.e., $t = \perp$. As the words in $win_{t,d',\rho}$ have an influence value based on Eq. (1), we can build a vector $\delta(win_{d',t,\rho})$ of length n. The entries $\delta(win_{d',t,\rho})[w]$ are set to 0 for each word $w \in \mathcal{V}$ not in $win_{t,d',\rho}$ and set to $I(w, win_{t,d',\rho)})$ otherwise. Using cosine similarity, the SCD t most similar to $\delta(win_{d',t,\rho})$ is given by:

$$\arg\max_{t} \frac{\delta(\mathcal{D})[t] \cdot \delta(win_{d',t,\rho})}{|\delta(\mathcal{D})[t]| \cdot |\delta(win_{d',t,\rho})|}. \tag{7}$$

Algorithm 2 describes estimating MPSCDs for d' using $\delta(\mathcal{D})$ given M. The output is the set of MPSCDs $g(d')$ as well as the windows and similarities for the MPSCDs in $g(d')$. The outer loop iterates over the positions of the M SCDs, setting up a window $win_{d',t,\rho}$ and a vector representation $\delta(win_{d',t,\rho})$. Then, Alg. 2 calculates cosine similarities between $\delta(win_{d',t,\rho})$ and the SCD vectors in $\delta(\mathcal{D})$ based on Eq. (7). It retains the SCD with the highest similarity as MPSCD t for the window $win_{d',t,\rho}$. Our approach rests on the following proposition:

Proposition 1. *Algorithm 2 estimates for a new document d' M (locally) most probable SCDs, i.e., Eq.* (7) *calculates for each window estimates of Eq.* (6)*.*

We argue that the similarity between the influence distribution over the words in a window and the SCD-word distribution indicates that the SCD is most likely to generate the words in the window. Another SCD generating other words with high probability would not generate the words in the window with a high probability and as such, does not lead to a high similarity.

The MPSCDs represent a local optimum based on the current setting of the windows. If the agent decides on adding the new document to the corpus and using the MPSCDs for additional tasks like query answering or document retrieval, optimizing the initial SCDs of d' might lead to more attuned SCDs.

4.4 Special Cases

We look at two special cases, one regarding fine-tuning MPSCDs, e.g., if interested in retaining SCDs for new documents, and one regarding large corpora.

SCD Window Adjustments. To optimize MPSCDs for a new document d', one can adjust the initial number of SCDs, the corresponding SCD positions, and the window sizes in d' to get MPSCDs with higher overall probability (or similarity).

We require the outputs of Alg. 2, (repository $g(d')$ for document d', set $\mathcal{W}$ containing the similarities sim and the initial windows $win_{d',t,\rho}$). To optimize

Algorithm 2 Estimating MPSCDs

```
1: function ESTIMATEMPSCD(Document $d'$, Number $M$, matrix $\delta(\mathcal{D})$)
2:     $\sigma \leftarrow \frac{words(d')}{M}$, $\rho \leftarrow \frac{\sigma}{2}$, $\mathcal{W} \leftarrow \emptyset$
3:     for $\rho \leftarrow \frac{\sigma}{2}$; $\rho \leq words(d)$; $\rho += \sigma$ do
4:         Set up a window $win_{d',t,\rho}$ of size $\sigma$ around $\rho$ with $t = \perp$
5:         $\delta(win_{d',t,\rho}) \leftarrow$ new zero-vector of length $n$
6:         for $w \in win_{d',t,\rho}$ do
7:             $\delta(win_{d',t,\rho})[w] += I(w, win_{d',t,\rho})$
8:         $t \leftarrow \arg\max_{t_i} \frac{\delta(\mathcal{D})[i] \cdot \delta(win_{d',t,\rho})}{|\delta(\mathcal{D})[i]| \cdot |\delta(win_{d',t,\rho})|}$ in $win_{d',t,\rho}$
9:         $sim \leftarrow \max_{t_i} \frac{\delta(\mathcal{D})[i] \cdot \delta(win_{d',t,\rho})}{|\delta(\mathcal{D})[i]| \cdot |\delta(win_{d',t,\rho})|}$
10:        $\mathcal{W} \leftarrow \mathcal{W} \cup \{(sim, win_{d',t,\rho})\}$
11:        $g(d') \leftarrow g(d') \cup \{t\}$
12:    return $g(d')$, $\mathcal{W}$
```

the SCDs in $g(d')$, we can iteratively adjust size and position of all windows in $\mathcal{W}$ and update t in each window based on Eq. (7) s.t. the overall similarity of the M influence vectors is maximum, i.e.:

$$\max \sum_{(sim, win_{d',t,\rho}) \in \mathcal{W}} sim. \tag{8}$$

Optimization starts with calculating the overall similarity of the current SCDs in $g(d')$ using the similarities stored in $\mathcal{W}$. Then, it iteratively adjusts windows until it reaches a local optimum for the optimization problem stated in Eq. (8). Each window in $\mathcal{W}$ can be adjusted in the following four different directions: (a) extend left boundary of the SCD-window to the left,(b) extend right boundary of SCD-window to the right,(c) shift left boundary of SCD-window to the right, and (d) shift right boundary of SCD-window to the left. Window adjustments (a) and (b) extend the size of window $win_{d',t,\rho}$, while window adjustments (c) and (d) reduce the size of window $win_{d',t,\rho}$.

Document clusters. Large corpora contain thousands of documents that may focus on different topics. Identifying documents in a corpus $\mathcal{D}$ having a high topic similarity with a new document d' narrows down the set of possible SCDs for d' by considering only SCDs from similar topics. Therefore, one can form a cluster $\mathcal{C}_{d'}$ of d'-related documents s.t. all $d \in \mathcal{C}_{d'}$ have a Hellinger distance H of their topic distributions θ being smaller than a threshold τ, i.e., $\mathcal{C}_{d'} = \{d \in \mathcal{D} \mid H(\theta_{d'}, \theta_d) < \tau\}$,where $H(\theta_d, \theta_{d'})$ refers to the Hellinger distance between the topic distributions of d and d', respectively. The cluster $\mathcal{C}_{d'}$ takes the place of $\mathcal{D}$ in Eq. (5). The threshold τ decides on the minimum required topic similarity between two documents d' and d, such that $d \in \mathcal{C}_{d'}$. The best value for τ depends on the performance of external applications working with the SCDs in $g(d')$. A smaller τ means a higher similarity between documents in $\mathcal{C}_{d'}$.

Next, we build MPSCDs into a decision making procedure for an agent.

4.5 Decision Making

An agent can use the MPSCDs for a new document d' making a decision on adding the new document d' to the corpus or not. Obviously, this decision depends on the goal of the agent. If the agent is interested in similar documents containing new information, it may decide adding documents where half of the SCDs have a high similarity and the other half of the SCDs have a small similarity. Based on the goal of the agent, thresholds have to be set accordingly.

The agent has to decide whether to extend its corpus $\mathcal{D}$ with d' if d' adds value relevant to its task. Document d' may not add value if d' has no or very little connection to $\mathcal{D}$ or if d' does not add new content. SCDs are the connection between documents and context, signalling their value to the task. Thus, the decision incorporates SCDs and the words that are connected to them using the available resources $\mathcal{D}$, $g(\mathcal{D})$, and $\delta(\mathcal{D})$. An agent proceeds as follows if presented with a new document d': (i) Using Algorithm Alg. 2, it computes MPSCDs $\{t_1, \ldots, t_M\}$ for d'. As a byproduct, it receives the windows and the maximum similarities of each window in $\mathcal{W}$. (ii) Based on $\mathcal{W}$ and $\{t_1, \ldots, t_M\}$, the agent makes a decision. The decision making is based on a combination of (i) maximum similarity, (ii) mininum similarity, (iii) average similarity, and (iv) maximum and average difference in the similarity between neighbouring windows.

Next, we look at a case study to test our multi-dimensional decision making.

5 Case Study

We present a case study illustrating the potential of the multi-dimensional decision making approach considering the initial question: Should an agent extend its reference library with a new document or should it not? In this case study we use two corpora containing Wikipedia articles. The first corpus contains documents about European largest cities (https://bit.ly/2kOvmwD); the second corpus contains documents about U.S. presidents (https://bit.ly/2Z1v1G9).

We compare for both corpora our multi-dimensional decision making approach with LDA. We do not consider entity matching since entity matching ignores context, which means that unrelated documents can share the same entities and similar documents can have no matches in their entities.

The new documents we test belong to the following four types of documents: (i) Similar documents (d_{sim}); content of new document is very similar to the content of documents in the corpus, e.g., the new document tells about the same event. (ii) Extensions (d_{ext}); content of new document is similar to the content of documents in the corpus and contains additional *information* unavailable in any other document in the corpus, e.g., the new document represents an extension of an article. (iii) Revisions (d_{rev}); the new document represents a revision of another document in the corpus. (iv) Unrelated documents (d_{unrel}); content of new document is unrelated to the content of documents in the corpus.

We preprocess all documents by lowercasing characters, stemming words, tokenizing, and eliminating tokens part of the Stanford CoreNLP stop-word list. Afterwards, we use Stanford OpenIE [1] to automatically extract tuples from the

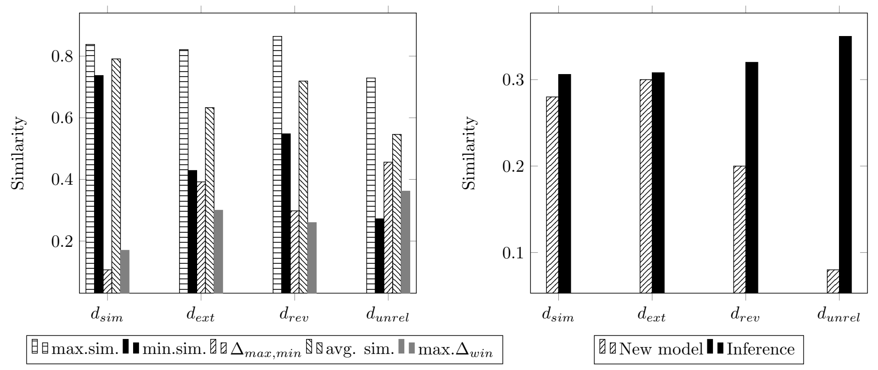

Fig. 1: Representation of the four document types d_{sim}, d_{ext}, d_{rev}, and d_{unrel} using the introduced MPSCD similarity values (left) and topic similarity (right).

documents acting as SCDs for all documents within the corpus to have roughly the same number of SCDs for all documents that are not influenced by us.

The MPSCDs of new documents support agents in their decision making process since MPSCDs of new documents give information about the document type. The MPSCDs are not only suitable to decide if a new document is similar or unrelated to the documents in the corpus, but also if a new document is an extension or a revision of a document in the corpus. We consider the following five indicators to decide on extending a corpus with a new document or not: (i) maximum similarity of all MPSCDs (max. sim.), (ii) minimum similarity of all MPSCDs (min. sim.), (iii) maximum difference between highest and lowest similarity value ($\Delta_{max,min}$), (iv) average MPSCD similarity (avg. sim.), and (v) maximum change between neighbouring MPSCD-windows (max. Δ_{win}).

On the left, Fig. 1 presents the indicators for the MPSCDs for each type of document. On the right, Fig. 1 shows two variants of comparing a new document and the corpus using LDA. The first variant learns a new topic model for the corpus including the new document, yielding a topic distribution for the new document. The second variant infers the topic distribution of the new document using an available topic model of the corpus. Similarity is given by the Hellinger distance between two topic distributions subtracted from the value 1. Infering the topic distribution is significantly faster than calculating a new topic model but makes it impossible to identify the type of a new document since all similarity values are similar. Generating a new topic model allows for distinguishing an unrelated document and a revision from all types but makes it difficult to distinguish a similar document from an extension because both documents share nearly the same topic distribution leading to almost identical similarities.

Estimating the MPSCDs for a new document and analysing the five indicators enables classification of a new document in the context of a given corpus. Generally, the value of indicators slightly change with the corpus.

Table 1: Document Type Indicator Comparision

	city corpus				president corpus			
	d_{sim}	d_{ext}	d_{rev}	d_{unrel}	d_{sim}	d_{ext}	d_{rev}	d_{unrel}
Max Sim.	$+$	$+$	$+$	$\circ$	$+$	$+$	$+$	$\circ$
Min Sim.	$+$	$\circ$	$\circ$	$-$	$\circ$	$\circ$	$\circ$	$-$
$\Delta_{max,min}$	$-$	$\circ$	$-$	$\circ$	$-$	$\circ$	$-$	$\circ$
Avg. Sim.	$+$	$\circ$	$+$	$\circ$	$+$	$+$	$+$	$\circ$
Max.Δ_{win}	$-$	$\circ$	$-$	$\circ$	$-$	$\circ$	$-$	$\circ$

Table 1 represents the five indicators for all four document types of the city corpus containing articles the largest European cities and the president corpus containing articles about the U.S. presidents. We specify a high similarity $(+)$ for values between 0.7 and 1, an average degree of similarity $(\circ)$ for values between 0.3 and 0.7 and a low similarity $(-)$ for values below 0.3.

For both corpora new documents of type d_{sim}, d_{ext} and d_{rev} share high values for the maximum similarity. Similar documents (d_{sim}) have a noticeable higher minimum similarity than all other types of documents. Unrelated documents (d_{unrel}) have a smaller maximum similarity value compared all other types of documents and the minimum similarity is small. The maximum similarity change between neighbouring windows ($max.\Delta_{win}$) of document extensions (d_{ext}) is similar to unrelated documents while the maximum similarity change between neighbouring windows of revisions (d_{rev}) is comparable to similar documents.

6 Conclusion and Outlook

If an agent is presented with an unknown document, this paper enables it to answer the question "To extend or not to extend?" in a context-specific way. The decision behind the question has to consider how much added value a document provides within the context of the agent's task. SCDs capture the context and generate the words of documents in introduced model. For a new document, an algorithm allows for estimating MPSCDs based on existing SCDs in the spirit of "how much of the new document can the existing SCDs generate with high probability?" For feasibility, the algorithm uses similarity of vector representations of word distributions. The procedure for making a decision uses the estimated MPSCDs and their similarities. By combining indicators, an agent is able to make a decision about documents with varying value to add.

Future work includes modelling a window sequence with a hidden Markov model to find a most probable sequence of known and unknown segments of a document for decision making. Another aspect is finding a global optimum, trading off better results with more work. Currently, we are further analyzing patterns emerging between windows to extract them for transfer learning and investigate kernel methods to separate the different types of new documents.

References

1. Angeli, G., Premkumar, M.J.J., Manning, C.D.: Leveraging linguistic structure for open domain information extraction. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers. pp. 344–354 (2015)
2. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of Machine Learning Research **3**, 993–1022 (2003)
3. Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Jr., E.R.H., Mitchell, T.M.: Toward an architecture for never-ending language learning. In: Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010 (2010)
4. Collarana, D., Galkin, M., Ribón, I.T., Vidal, M., Lange, C., Auer, S.: MINTE: semantically integrating RDF graphs. In: Proceedings of the 7th International Conference on Web Intelligence, Mining and Semantics, WIMS 2017, Amantea, Italy, June 19-22, 2017. pp. 22:1–22:11 (2017)
5. Cunningham, H., Tablan, V., Roberts, A., Bontcheva, K.: Getting more out of biomedical documents with gate's full lifecycle open source text analytics. PLoS Computational Biology **9**(2) (2013)
6. Dong, X.L., Gabrilovich, E., Heitz, G., Horn, W., Murphy, K., Sun, S., Zhang, W.: From data fusion to knowledge fusion. PVLDB **7**(10), 881–892 (2014)
7. Getoor, L., Diehl, C.P.: Link mining: a survey. SIGKDD Explorations **7**(2), 3–12 (2005)
8. Kuhr, F., Witten, B., Möller, R.: Corpus-driven annotation enrichment. In: 13th IEEE International Conference on Semantic Computing, ICSC 2019, Newport Beach, CA, USA, January 30 - February 1, 2019. pp. 138–141 (2019)
9. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hellmann, S., Morsey, M., van Kleef, P., Auer, S., Bizer, C.: Dbpedia - A large-scale, multilingual knowledge base extracted from wikipedia. Semantic Web **6**(2) (2015)
10. Newcombe, H.B., Kennedy, J.M., Axford, S., James, A.P.: Automatic linkage of vital records. Science **130**(3381), 954–959 (1959)
11. Papantoniou, K., Tsatsaronis, G., Paliouras, G.: KDTA: automated knowledge-driven text annotation. In: Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD 2010, Barcelona, Spain, September 20-24, 2010, Proceedings, Part III. pp. 611–614 (2010)
12. Reuters, T.: Opencalais. Retrieved June **16** (2008)
13. Sparck Jones, K.: A statistical interpretation of term specificity and its application in retrieval. Journal of Documentation **28**(1), 11–21 (1972)
14. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007. pp. 697–706 (2007)
15. Tanya Braun, Felix Kuhr, Ralf Möller: Unsupervised text annotations. In: Formal and Cognitive Reasoning - Workshop at the 40th Annual German Conference on AI (KI-2017) (2017)
16. Yang, J., Zhang, Y., Li, L., Li, X.: YEDDA: A lightweight collaborative text span annotation tool. In: Proceedings of ACL 2018, Melbourne, Australia, July 15-20, 2018, System Demonstrations. pp. 31–36 (2018)