# Constructing and Maintaining Corpus-driven Annotations

Felix Kuhr
University of Lübeck
Institute of Information Systems
kuhr@ifis.uni-luebeck.de

Ralf Möller
University of Lübeck
Institute of Information Systems
moeller@ifis.uni-luebeck.de

***Abstract*—A reference library can be described as a corpus of an individual composition of documents containing related work of research, documents of favorite authors, or proceedings of a conference. The documents in the corpus may change over time; new documents extend the corpus while other documents are sorted out. A subset of documents may contain meaningful annotations describing their content while other documents contain only weakly annotations. Enriching documents with meaningful annotations is beneficial for the performance of applications like semantic search, content aggregation, automated relationship discovery, query answering and information retrieval. However, enriching and maintaining a document with meaningful annotations is non-trivial. Available (semi-) automatic annotation tools ignore the individual composition of documents in corpora by annotating documents with generic named-entity related data. In this paper, we present techniques for enriching and maintaining annotations for document-specific databases considering changes in the composition of documents.**

## I. Introduction

In linguistics annotations add additional data to documents, supporting humans, and machines to understand the semantic meaning of words in the document. The granularity of semantic annotations depends on the application and a single annotation may cover a word, a sentence, a paragraph, a document, or an entire corpus [10] and the degree to which *added value* is brought to a document by enriching the document with annotations depends on the benefit for applications like semantic search, aggregation of content, automated relationships discovery, Query-Answering (QA), Information Retrieval (IR), document retrieval (DR), and Knowledge Management (KM).

In recent years, systems have emerged using methods from the domain of Information Extraction (IE) [3] and statistical relational learning (SRL) [16] to extract data from the text of thousands of unstructured documents and derive graph databases (DBs), representing a symbolic content description using extractable entities and relations between entities. Some of the most known systems are DeepDive [28], DBpedia, NELL [13], YAGO [17], FRED [11], and KnowledgeVault [6]. These systems generate graph DBs containing entities and relations from all the documents. Annotating documents with data from graph DBs relates to the entity-linking problem that is a well studied field [5], where entities from documents are linked to entities of graphs. However, matching words in the text of documents to entities that are in a graph DB is difficult, and even if the documents contain named-entities, simply annotating documents with entity-related data from graph DBs ignores the documents' semantics and higher purpose in mind of people selecting documents for the corpus. This also applies for graph DB representing an ontology, because the higher purpose in mind of humans is not involved within the process.

Obviously, collecting documents is not an end in itself and the documents in a corpus might represent related work of research, documents of favourite authors, or selective proceedings of conferences. A subset of annotations of a document's annotation database (ADB) may add value to another document's ADB within the same corpus e.g., by increasing the performance in document retrieval. Let us assume that a person is searching for documents about *iterative algorithms* within a personal reference library using some keywords like 'iterative algorithm' or 'EM-like algorithm'. Generally, documents are in the set of relevant documents, if they contain the keywords. However, if a document contain a specific iterative algorithm and the document does not contain the words *iterative algorithm* or *EM-like algorithm*, then the document is not in the set of relevant documents. However, if the keywords are in the document's ADB because of the annotation enrichment process, then the document can be part of the relevant documents and is possibly useful for the person searching for iterative algorithms in an individual collection of documents. Thus, we are interested in enriching document-specific annotation databases with annotations from the ADBs of other documents within the same corpus instead of using data from external graph DBs.

In this paper, we present an approach to enrich sparse and weakly annotated documents with annotations of documents in the same corpus taking advantage of the higher purpose in mind of people individually selecting the documents in a corpus. We introduce techniques for maintaining the annotations in annotation databases of documents within the same corpus handling new document extending the corpus, and documents dropping out of the corpus.

The remainder of this paper is structured as follows. Section II and III present related work and background information, respectively. In Section IV we introduce corpus-driven annotation enrichment of documents' ADBs. In Section V we conclude and present future work.

## II. Related Work

Over the recent years, a considerable number of automatic annotation systems have been introduced in the natural language processing (NLP) community. Automatic annotation systems use human language to directly extract data from the text of documents. A well-established technique is named-entity recognition (NER), which is a subtask of IE taking an unannotated block of text and producing an annotated block of text that highlights the names of entities and classify them into predefined categories such as persons, organizations, locations, etc. Some annotation systems extract named-entities from the text and use available DBs to identify more entities having a relationship to the extracted entities by using link prediction [12], which is the discipline of estimating the likelihood of the existence of a link between nodes, using the given links and attributes of nodes within a graph [19]. The granularity of annotations depends on the application and a single annotation may cover a word, a sentence, a paragraph, a document, or an entire corpus [10].

MINTE [2] is an approach for semantically integrating RDF graphs. This requires the management of data to determine the relatedness of different RDF representations of the same entity. Tipalo [9] is an automatic system identifying types from the text of Wikipedia documents for DBpedia entities. OpenCalais [15] is a knowledge extraction tool by Reuters, which automatically tags data in unstructured text using a large ontology. SemTag is a module of Seeker, both introduced by Dill et al. in [21] to generate semantic annotations. SemTag uses a structural analysis of text and the ontology *TAP* to automatically annotate documents with data from the ontology. Analogous to SemTag, KIM is a semantic annotation, indexing, and retrieval platform, developed by Ontotext [22] that identifies entities in the text of documents and links the entities to semantic descriptions which are provided by the KIMBO ontology (pre-populated ontology with many instances). The platform allows KIM-based applications to use it for automatic semantic annotation. KIM has been applied in different domains like anti-corruption and asset recovery, analysis of bio-medical content, or scientific papers. BOEMIE [14] is another approach, focusing on text block locations that correspond to specific types of named-entities, and additionally performs annotations of text that refers to the same topic to automatically creating annotations. For further annotation systems please refer to the survey of from Oliveira et al. [8].

Generally, all available annotation systems ignore the composition of documents and simply add data from external sources to documents to describe entities occurring in the documents. However, we believe that identifying DBs that are useful for annotating documents is as difficult as identifying a human expert adding high added value annotations to documents. Even if external DBs are available, we take the view that separately annotating documents simply adds domain-specific data to documents instead of describing the content of documents with respect to the document composition. Compared with existing automatic annotation systems, the contributions of this paper are:

- Constructing the ADBs of documents by considering the composition of documents by using an iterative algorithm.
- Maintaining the ADBs of documents by handling new documents extending the corpus as well as old documents dropping out of the corpus.

## III. Preliminaries

*Topic modeling techniques* estimate topics from a collection of documents and calculate for each of the documents a topic probability distribution $\theta$. Topics represent co-occurring words of the documents. The statistical technique called latent Dirichlet allocation (LDA)[20] generates a topic model from a set of documents to identify latent structures such as the topic distribution of documents and word topic distribution. This technique is identical to probabilistic latent semantic analysis (pLSA), except that the authors have introduced a sparse Dirichlet prior for the topic distribution expressing that documents cover only a small set of topics and that each topic uses only a small set of words frequently. It assumes documents to represent a mixture of topics where each topic is characterized by a distribution of words from a vocabulary containing all words of the collected documents. LDA uses a bag of words approach simplifying documents. For document $d$, LDA learns a discrete probability distribution $\theta_d$ that contains for each topic $k \in \{1, \ldots, K\}$ a value between 0 and 1. The sum over all $K$ topics for $d$ is 1. To find topically similar documents we use the Hellinger distance [27] measuring the distance between two probability distributions. Given two topic distributions $\theta_{d_i}$ and $\theta_{d_j}$ for documents $d_i$ and $d_j$, the Hellinger distance $H(\theta_{d_i}, \theta_{d_j})$ is given by $\frac{1}{\sqrt{2}}\sqrt{\sum_{k=1}^{K}\left(\sqrt{\theta_{d_i,k}} - \sqrt{\theta_{d_j,k}}\right)^2}$ where $k$ refers to the topics in the documents. Topic modelling techniques reduce the dimensionality of each document to the number of topics $k$. Having the topic model for the documents within corpus $\mathcal{D}$, it is feasible to calculate the Hellinger distance between documents $d_i$ and $d_j$. The result is a value between 0 and 1 and $H(\theta_{d_i}, \theta_{d_i}) = 0$. LDA has input documents $d_i$, $i \in \{1, ..., D\}$, where each document $d_i$ contains words $w_n$; $n \in \{1, ..., N\}$. The per-word topic assignment $z_{d,n}$ is drawn from a per-document topic distribution vector $\theta_d$. Each topic $k \in \{1, ..., K\}$ is a multinomial distribution of words $w$. LDA contains two hyperparameters $\alpha$ and $\beta$, where $\alpha$ conditions the per-document topic distributions $\theta_d$ and $\beta$ conditions the per-corpus topic distributions $\phi_k$, $k \in \{1, ..., K\}$.

*Information extraction* is a subdomain of NLP referring to methods that extract entities and their relations from text documents. Two main tasks of IE systems are NER and relation extraction. A possible result of an IE system is a set of Resource Description Framework (RDF) triples containing the extractable relations between entities. Identifying entities and relations within arbitrary long sentences containing subordinate clauses and other grammatical structures make IE

difficult. Some well known systems are OpenIE [3], Gate [7], and the framework document spanners [4]. We use OpenIE which learns a classifier to split sentences of text documents into shorter utterances and apply natural logic [24] to further shorten the utterances in a way such that the shortened utterances can be mapped to OpenIE triples representing subject, predicate, and object.

## IV. Corpus-driven Annotation Enrichment

In this section, we present the annotation enrichment process constructing ADBs of documents with annotations of related documents in the same corpus. We assume annotations of one document brining add value to another document in the same corpus, if the content of both documents is somehow related. Thus, enriching ADB $g_e$ with annotations from $d_e$-related documents requires the identification of $d_e$-related documents and those annotations that are semantically related to the annotations of document $d_e$. Kuhr et al. [1] have introduced two similarities namely D- and G-similarity identifying $d_e$-related documents in corpus $\mathcal{D}$ and have presented the iterative Algorithm 1 using the two similarity measures identifying $d_e$-related documents to assign each annotation with an Expected Relevance Value (ERV) to identify the annotations describing the content of $d_e$ without focusing on named-entities.

The D-similarity is based on the idea of topic modelling and compares the relatedness between two documents using the similarity of the documents' topics. The document-specific topic vector is known as the topic distribution of a document. D-Similarity is defined by:

$$Sim_D(d_e, d_k) = 1 - H(\theta_{d_e}, \theta_{d_k}), \tag{1}$$

where $H(\theta_{d_e}, \theta_{d_k})$ estimates the Hellinger distance between the topic distributions of $d_e$ and $d_k$ and $Sim_D(d_e, d_k) \in [0, 1]$. The interval follows directly from the definition of the Hellinger distance. The higher the D-similarity the more similar the documents' topic distribution. The text of documents $d_e$ and $d_k$ having a high D-similarity contain similar content such that annotations for $d_k$ might be added value for $d_e$. The similarity function $s(g_e^i, g_k^j)$ calculating a similarity score between two annotations, comparing the i-th annotation in $g_e$ with the j-th annotation in $g_k$ using the entities and relations to estimate a similarity score in $[0, 1]$. The more similar two annotations $g_e^i \in g_e$ and $g_k^j \in g_k$ the higher $s(g_e^i, g_k^j) \in [0, 1]$ and function $s(g_e^i, g_k^j)$ is defined by:

$$s(g_e^i, g_k^j) = \begin{cases} 0 & \text{if}(s^i \neq s^j \wedge p^i \neq p^j \wedge o^i \neq o^j) \\ \frac{1}{3} & \text{if } (s^i = s^j \wedge p^i \neq p^j \wedge o^i \neq o^j) \vee \\ & (s^i \neq s^j \wedge p^i = p^j \wedge o^i \neq o^j) \vee \\ & (s^i \neq s^j \wedge p^i \neq p^j \wedge o^i = o^j), \\ \frac{2}{3} & \text{if } (s^i = s^j \wedge p^i = p^j \wedge o^i \neq o^j) \vee \\ & (s^i \neq s^j \wedge p^i = p^j \wedge o^i = o^j) \vee \\ & (s^i = s^j \wedge p^i \neq p^j \wedge o^i = o^j), \\ \frac{3}{3} & \text{if } (s^i = s^j \wedge p^i = p^j \wedge o^i = p^j), \end{cases} \tag{2}$$

where variable $s$ represents the subject, $p$ is the predicate, and $o$ the object of an annotation.

The G-similarity identifies $d_e$-related documents in $\mathcal{D}$ comparing annotations of $g_e$ with annotations of other documents' ADB and is defined as

$$Sim_G(g_e, g_k) = \frac{1}{2} \cdot \overline{v^c} + \overline{v^r}, \tag{3}$$

where $\overline{v^c}$ and $\overline{v^r}$ represents the average value of the similarity vectors taking the ratio between high and low similarity scores into account such that two ADBs $g_e$ and $g_k$ sharing only a small number of high similarity scores and a high number of low similarity scores have a small G-similarity. $v^c \in \mathbb{R}^n$, with $v_j^c = max_i\ a_{i,j}$ represents the similarity vector containing for each annotation in $g_e$ the highest possible similarity score and $v^r \in \mathbb{R}^m$, with $v_i^r = max_j\ a_{i,j}$ represents the similarity vector containing for each annotation in $g_k$ the highest possible similarity score.

### A. Iterative Annotation Construction

Next we present the iterative annotation enrichment Algorithm 1, which is based on Dempster et al. [26]. Their EM-algorithm estimates the maximum likelihood of parameters handling unobserved variables alternating between the expectation and maximization step. The expectation step creates a function for the expectation of log-likelihood using the present values for the parameters. The maximization step calculates the parameters maximizing the expected log-likelihood in the expectation step. The expectation step of Algorithm 1 identifies $d_e$-related documents, represented as $\mathcal{D}^{d_e}$, using D- and G-similarity and calculates for all annotations $\mathcal{G}^{d_e}$ the ERV value. The maximization step calculates the new average G-similarity optimizing the ERVs in the next expectation step. The ERV estimates only the annotations in $\mathcal{G}^{d_e}$ describing the semantic meaning of the content from document $d_e$ as

$$ERV_t^{d_e} = \overline{Sim_{D_t}} \cdot \overline{Sim_{G_t}} \cdot f(t), \tag{4}$$

where $\overline{Sim_{D_t}}$ is the average D-similarity of documents $d \in \mathcal{D}^{d_e}$ such that $g$ contains annotation $t$, $\overline{Sim_{G_t}}$ is the average G-similarity of all ADBs $g \in \mathcal{G}^{d_e}$ containing annotation $t$ and $f(t)$ is the frequency of $g \in \mathcal{G}^{d_e}$ containing annotation $t$. The average D-similarity of documents where the corresponding ADBs contain annotation $t$ is given by $\overline{Sim_{D_t^{d_e}}}$. $\overline{Sim_{G_t^{d_e}}}$ represents the average G-similarity containing annotation $t$ and $\overline{ERV^{d_e}}$ is the average ERV of all annotations in $d_e$. There are two ways leading to a high ERV. First, D- and G-similarity between $d_e$ and $\mathcal{D}^{d_e}$ is high which means the text of each $d \in \mathcal{D}^{d_e}$ and $d_e$ is semantically related. Second, the number of documents in $\mathcal{D}^{d_e}$ containing annotation $t$ is high. Thus, enriching ADB of $d_e$ with annotation $t$ occurring in many other ADB may add value to the ADB of $d$, because it seems to be generic or very specific for those documents. The input parameters of Algorithm 1 are document $d_e$, $g_e$, $\mathcal{D} \setminus \{d_e\}$, and D-similarity selection threshold $\tau$. The output is the optimal ADB $g'_e$. In the E-Step, the algorithm updates variable $erv_t$ for each annotation $t$ in $\mathcal{G}_e^d$. The algorithm adds annotations with

**Algorithm 1** Iterative Annotation Enrichment

1: **Input**: $d_e$, $g_e$, $\mathcal{D} \setminus \{d_e\}$, $\tau$
2: **Output**: $g'_e$
3: **Define:** $\epsilon = 0.1$ , $\mathcal{D}^{d_e}$, $\mathcal{D}'^{d_e}$, $\mathcal{G}^{d_e}$, $g'_e$
4: **Initialize:** $\overline{Sim_{G^{d_e}}} = \epsilon$, $\overline{Sim'_G} = \overline{Sim_{G^{d_e}}} - \epsilon$, $\mathcal{D}^{d_e} = \emptyset$, $\mathcal{G}^{d_e} = \emptyset$, $erv_t^{d_e} = 0$, $g'_e = \emptyset$
5: **while** $|\overline{Sim_{G^{d_e}}} - \overline{Sim'_{G^{d_e}}}| \geq \epsilon$ **and** $\overline{Sim_{G^{d_e}}} > \overline{Sim'_G}$ **do**
6: $g'_e \leftarrow g_e$
7: $\mathcal{D}^{d_e} \leftarrow \emptyset$ ▷ E-Step
8: **for each** $d_k \in \mathcal{D}$ **do**
9: **if** $Sim_D(d_e, d_k) > \tau$ **and** $Sim_G(g'_e, g_k) > \overline{Sim_{G^{d_e}}}$ **then**
10: $\mathcal{D}^{d_e} \leftarrow \mathcal{D}^{d_e} \cup \{d_k\}$
11: **end if**
12: **end for**
13: **for each** $t \in \mathcal{G}^{d_e}$ **do**
14: $erv_t^{d_e} \leftarrow erv_t^{d_e} + ERV_t^{d_e}$
15: **end for**
16: **for each** $t \in \mathcal{G}^{d_e}$ **do**
17: **if** $ERV_t^{d_e} > \overline{ERV}^{d_e}$ **then**
18: $g'_e \leftarrow g'_e \cup \{t\}$
19: **end if**
20: **end for** ▷ M-Step
21: $\overline{Sim'_{G^{d_e}}} = \overline{Sim_{G^{d_e}}}$
22: $\overline{Sim_{G^{d_e}}} = \frac{\sum_{k=1}^{|\mathcal{D}^{d_e}|} Sim_{G^{d_e}(g_e, g_k)}}{|\mathcal{D}^{d_e}|}$
23: **end while**
24: **return** $g'_e$

high ERV to ADBs and ignores annotations with low ERV. In the M-Step, the algorithm updates the average G-similarity $\overline{Sim_{G^{d_e}}}$ which is part of the termination condition in line 5.

We use Algorithm 1 constructing for each document within corpus $\mathcal{D}$ the corresponding corpus-driven ADB by using annotations of related documents' ADB.

### *B. Maintaining Annotation Databases*

Over time, the composition of documents might change for various reasons; e.g., a subset of documents drops out of the corpus or new documents extend the corpus. We describe first ideas for techniques handling adjustments in the composition of document of the corpus for maintaining the annotations in the document-specific ADBs. First, we describe techniques maintaining the annotations in ADB of documents while new documents extending the given set of documents in a corpus. Afterwards, we describe techniques maintaining the annotations of documents' ADBs while documents are dropping out of the corpus.

*1) Corpus Extending Documents.:* Documents in a corpus are not fixed and sometimes new documents are available containing text which is relating to the content of other documents in the corpus, so that a person might extend the corpus with new documents. Extending the corpus with a new, unseen document nothing is available, except from the text of the new document. Enriching a new document with annotations from other documents in the corpus requires the identification of similar documents and similar annotation database, such that Algorithm 1 can compare the documents in the corpus with the new document using the D- and G-similarity to construct the ADB of a new document with annotations of other documents having both, high D- and G-similarity. But, how to identify documents within the corpus having a high D-similarity with the new document; however, when only the text is available for the new document?

The D-similarity estimates the similarity between two documents using their topic distributions. Therefore, it is required to estimate the topic distribution for the new document. Generally, documents' topic distribution rest upon the topic-word distribution derived from the text of all documents within the corpus and it is impossible to compare a new document with other documents from a corpus using the D-similarity without estimating the topic distribution for the new document considering the documents in the corpus. There are two approaches estimating the topic distribution for a new, unseen document: i) Extending the corpus with the new document and calculating a new topic model from the text of all documents in the *new*, extended version of the corpus. ii) Approximating the topic distribution of the new document by inferring the new document's topic distribution from the two corpus-specific distributions; namely, the topic-word distribution and the document-topic distribution. The topic distribution for each of the old documents is unchanged. Estimating a new topic model from a given set of documents in a corpus $\mathcal{D}$ is computation-intensive in comparison to approximating the topic distribution for a single new document using the data from a given topic model. Hence, it is a debatable point whether it is necessary to calculate a new topic distribution for all documents in $\mathcal{D}$, just because a new document $d'$ extends corpus $\mathcal{D}$ to $\mathcal{D}' = \mathcal{D} \cup \{d'\}$, instead of approximating the topic distribution for the new document $d'$ using available data from the initial topic model generated from the documents in corpus $\mathcal{D}$. It is possible to infer the topic distribution for a new document $d'$ by using the parameters of the topic model generated from all documents in $\mathcal{D}$. The *folding in* Gibbs sampling technique [23] which is the same as Gibbs sampling [25], except the sampling uses the topic-word distribution $\phi$, and per document-topic distribution $\theta$ of the topic model generated from the documents in $\mathcal{D}$ approximates the topic distribution for $d'$.

In a *folding in* Gibbs sampling approach we can estimate for each word $w$ in the new document $d'$ the most probable topic which is initialized using $\phi$. If $d'$ contains a new word $w$ not part of any document $d \in \mathcal{D}$, we randomly assign the topic for this word. The *folding in* Gibbs sampling estimates only the topic assignment for each word in the new document $d'$. The topic assignment of the words in $d'$ gives the distribution of topics in $d'$. However, the documents' topic distribution changes with the documents in corpus $\mathcal{D}$ and it is recommended to estimate a new topic model after extending $\mathcal{D}$ with new documents. There is no limitation in the number of documents we can add to the corpus without estimating a new topic model from all documents in $\mathcal{D}'$, but the initial topic model ignores the content of all new documents. Hence, it might be necessary to generate a new topic model after a while. The number of documents in the corpus, length of

the documents and words within documents is responsible for the error between the approximation using *folding in* Gibbs sampling and generating a new topic model using the extend corpus. Both techniques lead to a topic distribution for new, unseen documents and having the topic distribution for new documents it is possible to compare the new documents with all documents in the corpus using the D-similarity.
However, Algorithm 1 compares the similarity between documents using the G-similarity besides the D-similarity; but how to compare the annotations of documents in $\mathcal{D}$ with a new document having no annotations? i) We have to identify extractable annotations from the text of the document using a combination of IE techniques. ii) After extracting the annotations and store them in the document's ADB, the ADB consists of annotations representing at least a subset of the document's content.

The initial set of annotations is small and contains only data directly extractable from the text; the quality of annotations highly depends on the extraction techniques. However, the quality of annotations is not a problem as we are interested in constructing and maintaining the ADB of a new document. Initially, only a small set of annotations in the document's ADB is required, describing parts of the document's content to calculate the G-similarity working at annotation-level. Algorithm 1 is capable of iteratively enriching the ADB of the new document $d'$ with annotations of documents within the same corpus, having both, a high D- and G-similarity to $d'$.

*2) Dropping out Documents.:* Similar to the documents extending a corpus it is possible that documents may drop out of the corpus for various reasons; e.g., the content of a document is outdated or contain wrong information such that the document is no longer relevant for the library of a person. As mentioned aforetime, each document has a link to a document-specific ADB containing annotations for the document. Within the iterative annotation enriching process, Algorithm 1 constructs the ADBs of documents using a subset of annotations from similar documents' ADBs of the corpus. If a document $d$ is dropping out of corpus $\mathcal{D}$, we have to decide how to deal with the ADBs of other documents within corpus $\mathcal{D}$ containing annotations from ADB $g$ of document $d$, since Algorithm 1 enriched the ADBs of documents with annotations from $d$ dropping out of the corpus.

Generally, there are three approaches for handling the annotations of documents dropping out of the corpus: 1) *No rollback*: Do not change the remaining documents' ADBs, just because the ADBs contain annotations being enriched from a document's ADB no longer part of the corpus. Let us assume that a person removes document $d$ from corpus $\mathcal{D}$. The reason for removing document $d$ from $\mathcal{D}$ is that a new, extend version of document $d$ exists, represented by $d'$, and we would like to replace $d$ by $d'$. Then there is no reason for removing the annotations in documents' ADB being enriched from $g$ of $d$. 2) *Partly rollback*:If the content of a document $d$ is outdated and the document is no longer part of the corpus, it might be desired to remove the initial annotations in $g$ from all ADBs being enriched with initial annotations of $g$. 3) *Rollback*: Remove all iteratively enriched annotations from all documents' ADB, generate a new topic model from the documents in the reduced corpus and perform Algorithm 1 to enrich the ADBs using annotations of other documents' ADB.

Simply removing all annotations of a dropping out document's ADB from the ADBs of all remaining documents is an approximation for the documents' annotations, and it is not possible to track the origin of annotations. It might be possible that an ADB $g_e$ is similar to another ADB $g_i$, just because $g_e$ contains an annotation from a documents' ADB not in the corpus anymore. Furthermore, Algorithm 1 enriches $g_e$ with annotations from $g_i$. By simply removing all initial annotations of a dropping out document's ADB from all other documents' ADBs ignore the iteratively enriched annotations by a high G-similarity. Thus, another approach is removing all iteratively enriched annotations from the ADBs and performs Algorithm 1 for all documents again. This approach is more expensive than simply removing only all initial annotations of a dropping out document's ADB from all other documents' ADBs. However, we assume that the document-specific annotations better describe the higher purpose in mind of the person collecting the documents in the corpus. Generating a new topic model from the remaining documents in the reduced corpus and performing Algorithm 1 to enrich the ADBs using annotations of other documents' ADB has the advantage in only adding annotations to documents' ADB having a high D- and G-similarity. However, as already mentioned previously, calculating a new topic model and performing Algorithm 1 on all documents is computationally intensive. Depending on the number of documents dropping out of the corpus approximating documents' annotations might be good enough.

## V. Conclusion

In this paper, we have presented the construction of document-specific ADBs using two holistic similarities called D- and G-similarity and the iterative EM-like Algorithm 1. Additionally, we have introduced different approaches for maintaining annotations in annotation database since the composition of documents might change over time; new documents extend the corpus and old documents are dropping out.

In future work we are interested in estimating parameters to decide which maintaining approach fits best to a given collection of documents such that the required resources for maintaining documents ADB is as small as possible.

## References


[1] Felix Kuhr, Bjarne Witten, and Ralf Möller. "Corpus-driven Annotation Enrichment". In: *13th IEEE International Conference on Semantic Computing*. IEEE Computer Society, 2019.

[2] Diego Collarana et al. "MINTE: semantically integrating RDF graphs". In: *Proceedings of the 7th International Conference on Web Intelligence, Mining and Semantics*. ACM. 2017, p. 22.

[3] Gabor Angeli, Melvin Johnson Premkumar, and Christopher D Manning. "Leveraging linguistic structure for open domain information extraction". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL 2015)*. 2015.

[4] Ronald Fagin et al. "Document spanners: A formal approach to information extraction". In: *Journal of the ACM (JACM)* 62.2 (2015), p. 12.

[5] Wei Shen, Jianyong Wang, and Jiawei Han. "Entity linking with a knowledge base: Issues, techniques, and solutions". In: *IEEE Transactions on Knowledge and Data Engineering* 27.2 (2015), pp. 443–460.

[6] Dong, Xin Luna and Gabrilovich, Evgeniy and Heitz, Geremy and Horn, Wilko and Murphy, Kevin and Sun, Shaohua and Zhang, Wei. "From data fusion to knowledge fusion". In: *Proceedings of the VLDB Endowment* 7.10 (2014), pp. 881–892.

[7] Hamish Cunningham et al. "Getting more out of biomedical documents with GATE's full lifecycle open source text analytics". In: *PLoS computational biology* 9.2 (2013), e1002854.

[8] Pedro Oliveira and João Rocha. "Semantic annotation tools survey". In: *Computational Intelligence and Data Mining (CIDM), 2013 IEEE Symposium on*. IEEE. 2013, pp. 301–307.

[9] Aldo Gangemi et al. "Automatic typing of DBpedia entities". In: *International Semantic Web Conference*. Springer. 2012, pp. 65–81.

[10] ISO ISO. "24612: 2012 Language resource managementLinguistic annotation framework (LAF)". In: *Project leader: Nancy Ide* (2012).

[11] Valentina Presutti, Francesco Draicchio, and Aldo Gangemi. "Knowledge extraction based on discourse representation theory and linguistic frames". In: *International Conference on Knowledge Engineering and Knowledge Management*. Springer. 2012, pp. 114–129.

[12] Linyuan Lü and Tao Zhou. "Link prediction in complex networks: A survey". In: *Physica A: statistical mechanics and its applications* 390.6 (2011).

[13] Carlson, Andrew and Betteridge, Justin and Kisiel, Bryan and Settles, Burr and Hruschka Jr, Estevam R and Mitchell, Tom M. "Toward an Architecture for Never-Ending Language Learning." In: *AAAI*. Vol. 5. 2010.

[14] Pavlina Fragkou et al. "BOEMIE Ontology-Based Text Annotation Tool." In: *LREC*. Citeseer. 2008.

[15] Thomson Reuters. "OpenCalais". In: *Retrieved June* 16 (2008).

[16] Lise Getoor and Ben Taskar. *Introduction to statistical relational learning*. MIT press, 2007.

[17] Suchanek, Fabian M and Kasneci, Gjergji and Weikum, Gerhard. "Yago: a core of semantic knowledge". In: *Proceedings of the 16th international conference on World Wide Web*. ACM. 2007, pp. 697–706.

[18] Alexander Yates et al. "Textrunner: open information extraction on the web". In: *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*. Association for Computational Linguistics. 2007, pp. 25–26.

[19] Lise Getoor and Christopher P Diehl. "Link mining: a survey". In: *Acm Sigkdd Explorations Newsletter* 7.2 (2005), pp. 3–12.

[20] David M Blei, Andrew Y Ng, and Michael I Jordan. "Latent dirichlet allocation". In: *Journal of machine Learning research* 3.Jan (2003).

[21] Stephen Dill et al. "SemTag and Seeker: Bootstrapping the semantic web via automated semantic annotation". In: *Proceedings of the 12th international conference on World Wide Web*. ACM. 2003, pp. 178–186.

[22] Borislav Popov et al. "KIM–semantic annotation platform". In: *International Semantic Web Conference*. Springer. 2003, pp. 834–849.

[23] Andrew Kachites McCallum. "MALLET: A Machine Learning for Language Toolkit". http://mallet.cs.umass.edu. 2002.

[24] Víctor Manuel Sánchez Valencia. *Studies on natural logic and categorial grammar*. Universiteit van Amsterdam, 1991.

[25] Stuart Geman and Donald Geman. "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images". In: *IEEE Transactions on pattern analysis and machine intelligence* 6 (1984), pp. 721–741.

[26] Arthur P Dempster, Nan M Laird, and Donald B Rubin. "Maximum likelihood from incomplete data via the EM algorithm". In: *Journal of the royal statistical society. Series B (methodological)* (1977), pp. 1–38.

[27] Ernst Hellinger. "Neue Begründung der Theorie quadratischer Formen von unendlichvielen Veränderlichen." In: *Journal für die reine und angewandte Mathematik* 136 (1909), pp. 210–271.

[28] Ce Zhang. "DeepDive: a data management system for automatic knowledge base construction". PhD thesis. The University of Wisconsin-Madison.