Unsupervised Text Annotation

Tanya Braun, Felix Kuhr, and Ralf Möller

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

Abstract. We introduce the unsupervised text annotation model UTA, which iteratively populates a document-specific database containing the related symbolic content description. The model identifies the most related documents using the text of documents and the symbolic content description. UTA extends the database of one document with data from related documents without ignoring the precision.

Keywords: Unsupervised Text Annotation

1 Introduction

In recent years, knowledge graph systems have emerged using methods of information extraction (IE) and statistical relational learning (SRL) to extract data from documents and build knowledge graphs representing a symbolic content description in a graph using entities and relations.

Some of the most known knowledge graph systems are DeepDive [2], Never Ending Language Learner (NELL) [5], YAGO [6] and KnowledgeVault [3]. These systems generate knowledge graphs using documents and external sources available on the web without a specific user in mind. The systems often produce graphs with billions of edges linking millions of nodes and have some form of a supervision, either through interaction or prior information. The number of entities and relations lead to a huge graph such that the approach of these knowledge graph systems is usually not tailored towards special tasks other than the accumulation of data. Additionally, most of the systems only make use of the entities and relations but ignore the original text document after extracting entities and relations from the text.

Given the limitation that most knowledge graph systems produce huge graphs, we present the approach Unsupervised Text Annotation (UTA) focussing on document-specific data representation linking each document to a graph database (DB) where graphs represent entities and relations extractable from the linked text document. An iterative process extends the graph with entities and relations, which are not directly extractable from the document itself but extractable from the related document. An important aspect of the approach is that it does not ignore the text of the documents after extracting entities and relations, but uses the text as a supporting mechanism to decide if entities and relations are relevant for a document-specific content description.

UTA employs methods from IE and SRL to address the following problems: (i) Adding entities and relations to document specific DBs by hand is time-consuming. (ii) Association is complex as it has to add relevant entities and relations without adding irrelevant data to achieve high precision and recall. Precision and recall compare the document-specific DBs against the DBs built by human text annotators.

To handle the first problem, we use IE techniques to extract entities and relations for documents. For the second problem, we work on a small subset of documents to enrich a document specific DB using topic modeling techniques to identify related documents. Instead of simply adding entities and relations to the DB, we perform a text segment analysis of the text segments. Two documents, sharing the same entities, might contain content about two completely different subjects. Only if the two text segments are similar, the entities and relations of related documents extend the DB.

Thus, UTA contains an algorithm that first identifies for each document other related documents and second extracts entities and their relations from the documents to extend the document-specific DB with relevant data. To generate a document-specific DB, we need to (i) extract entities and corresponding relations from documents, (ii) perform localization of entities and relations within the text document, (iii) enrich a document-specific DB with data from related documents. We name data from related documents as associative data.

The remainder of this paper is structured as follows. We first introduce background information on topic modeling techniques and information extraction. We then present the algorithm to generate and extend document specific graphs. Next, we provide a case study to show the potential of the unsupervised text annotation approach. Last, we present a conclusion.

2 Preliminaries

This section presents latent Dirichlet allocation (LDA) topic model and brief information about IE.

Topic Modeling Techniques Topic modeling techniques learn topics from a collection of documents and calculate for each of the documents a topic probability distribution (aka *topic proportions*). Topics represent co-occurring words of the documents but have no high-level description, like *sports* or *film-industry.* LDA [7] is a well known generic topic model providing good results while being relatively simple. It uses a bag of words approach simplifying documents and topics depend only on a document's word frequency distribution. For a 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 similar documents we use the Hellinger distance [9] measuring the distance between two probability distributions. Given two topic proportions θ_{d_i} and θ_{d_j} for documents d_i and d_j, the Hellinger distance $H(\theta_{d_i}, \theta_{d_j})$ is given by $\sum_{k=1}^{K} \left(\sqrt{\theta_{d_i,k}} - \sqrt{\theta_{d_j,k}}\right)^2$ where k refers to the topics in the documents.

Information Extraction IE, a subdomain of natural language processing, refers to methods that extract entities and their relations from text. Two main tasks of an IE system are named-entity recognition and relation extraction. A possible outcome of an IE system is a set of Resource Description Framework (RDF) triples containing the extracted entities and relations for a document. Identifying entities and relations within arbitrary long sentences containing subordinate clauses and other grammatical structures makes the task difficult.

An example IE system is the OpenIE library from Stanford based on [1]. The IE system consists of two stages: (i) Learn a classifier to split sentences of text documents into shorter utterances. (ii) Apply natural logic to further shorten the utterances in a way such that the shortened utterances can be mapped to OpenIE triples representing subject, predicate, and object.

3 Unsupervised Text Annotation

This section presents a formal description of the unsupervised text annotation approach, which extracts entities and relations directly from a text document and enriches the DB with data from related documents. We use the term *local DB* to refer to a document-specific graph representing a content description. We distinguish between (i) *extractable data*, which is directly extractable from the text of a document (ii) *associative data*, which is part of another document's local DB, and (iii) *inferred data*, which is not part of any local DB and therefore, new at a global level. The overall objective of UTA is to iteratively enrich local DBs with data from other documents from the same document corpus.

Definition 1 (Document corpus): Let d be a document and $\{d_i\}_{i=0}^{n}$ be a set of n documents, then $\mathcal{D}$ is the document corpus representing the set $\{d_i\}_{i=0}^{n}$.

Each document d_i in $\mathcal{D}$ links to a *local DB* g_i, where g_i is a graph representing the entities and relations between the entities.

Definition 2 (Graph): A graph is a pair $g = (V, E)$ consisting of a set of V and E, where the elements of V are vertices, representing entities, and the elements of E are edges, representing relations between entities. Edge $e = (u, v, r, l)$ presents a relation r between vertices u and v and contains the localization l. The localization l refers to the text segment of the corresponding document containing the entities and relation.

Enriching a DB g_i of document d_i is the process of extending g_i with data from related documents. The *global DB* $\mathcal{G}$ of $\mathcal{D}$ is given by the union of all local DBs $\cup_{i \in \mathcal{D}} G_i$.

For a document $d_i \in \mathcal{D}$ and $g_i \in \emptyset$ adding data to g_i requires the following steps (i) remove d_i from $\mathcal{D}$, (ii) obtain entities and their relations from d_i using IE techniques, (iii) localization of entities and relations to the corresponding position in the text of d_i, (iv) identification of documents D^{d_i} which are related to d_i, (v) enrich g_i with data from G^{d_i}, (vi) return d_i to $\mathcal{D}$.

Corpus $\mathcal{D}$, and thereby global DB $\mathcal{G}$, is bound to become too large at some point. Thus, for enriching g_i, only a subset $D^d \subseteq \mathcal{D}$ of documents related to d_i is selected to enrich the DB g_i.

Algorithm 1 Iterative Database Construction

```
while true do
    $remove(\mathcal{D}, d_i)$
    if $g_i = \emptyset$ then
        $g_i \leftarrow IE(d_i)$
    end if
    $D^d \leftarrow select(\mathcal{D}, d)$
    $G^d \leftarrow \cup_{j \in \mathcal{D}^d} g_j$
    $g_i \leftarrow enrich(G_i^d, g_i)$
    $add(\mathcal{D}, d_i)$
end while
```

Algorithm 1 shows a pseudo code description of the iterative DB construction with procedures $select(\mathcal{D}, d)$ and $enrich(G^d, g)$ for selecting D^d, and enriching the DB g of document d with data from D^d. We reach a fix point where, unless we add new data through a new document, no changes in the local DBs occur.

The following sections detail about identifying related documents and enriching documents' DBs.

3.1 Document Selection

The selection of documents consists of two parts, (i) identifying similar documents for document d using matching techniques and (ii) building a set D^d from matching documents.

Document Matching Enriching d_i's DB g_i with data from other DBs g_j requires the identification of documents containing content matching the content of current document d_i with respect to the following aspects: (i) similarity in topic proportions, (ii) subgraph isomorphism between graph g_i and g_j.

Both matching techniques produces a set of documents, $D^{d,top}$ and $D^{d,iso}$. D^d represents the union of both, $D^{d,top}$ and $D^{d,iso}$. For both techniques, we specify assumptions and selection criteria.

The first technique, *similarity in topic proportions*, uses the text of documents. We assume that two documents sharing similar topic proportions contain content of related topics. Thus, one's DB might enrich the other's DB. Document d_j is part of $D^{d,top}$ if $H(\theta_d, \theta_{d_j}) < t_1$. Adding new documents to the corpus $\mathcal{D}$ changes the topics. Hence, an update of topics and topic proportions is required after the document corpus is extended with new documents.

The second matching technique performs comparison on the level of local DBs. We assume that related documents share entities and relations, i.e., have the same vertices or edges, and thus, contain information of interest for the other. We adopt a heuristic based on the Jaccard index to decide if a document d_j is relevant for another document d_i. The Jaccard index for two local DBs g_i and g_j is given by $J(g_i, g_j) = \frac{|g_i \cap g_j|}{|g_i \cup g_j|}$. We use J with DBs as inputs meaning, we compare DB elements for equality. For J, we exclude the text segment localization variable l from all edges and consider partial overlaps of vertices and edges within the graphs representing the DBs. Given two edges $e' := (u', v', r', l')$ and $e'' := (u'', v'', r'', l'')$, we say e' and e'' have a *full* overlap (and are equal) *iff* $u' = u''$, $v' = v''$, and $r' = r''$. A *partial* overlap occurs if a non-empty subset of vertices (entities) and edges (relations) overlap. We incorporate the different overlaps using weights. A full overlap receives the weight $w_3 = 1.0$, a two-entity overlap the weight $w_2 = 0.75$, a two-component overlap consisting of one entity and one relation receives the weights $w_1 = 0.5$, and a one-component overlap the weight $w_0 = 0.25$. Given two DBs g_i and g_j, let n_3, n_2, n_1, and n_0 denote the number of elements with a full, a two-entity, a two-component, and a one-component overlap respectively. We add document d_i to $D^{d_i,iso}$ if $J(g_i, g_j) = \frac{n}{|g_i|+|g_j|-n} > t_2$, $n = n_3 \cdot w_3 + n_2 \cdot w_2 + n_1 \cdot w_1 + n_0 \cdot w_0$. For g_j to have data to add to g_i, $g_j \supset g_i$ has to hold. Using the Jaccard index allows an identification of a set of documents in $D^{d,iso}$ which can enrich the DB of document d.

3.2 Database Enrichment

In the previous section, we have shown how to select documents D^d potentially enriching the DB g of d. This subsection explains how to enrich g using the DBs G^d belonging to the matching documents D^d and focussing on both, high precision and high recall, comparing the document-specific DBs against the DBs built by human text annotators.

Potential associative data for DB g of document d is the data in G^d that is not already in g. For adding associative data, we are interested in $G^d \setminus \{g\}$. Unfortunately, $G^d \setminus \{g\}$ may be very large if D^d is large. Simply adding all data from G^d not in g extends g to a large DB containing data not closely related to d. Consequently, the precision would be very low. Circumventing additions of irrelevant data to g requires filtering techniques. The following steps filter data in G^d: (i) Build a set G'^d of candidate data. Given G^d, candidate data e_{d_j} of document d_j is a relation r between two entities u and v, where relation r appears in at least n DBs of G^d or the text segment related to l of e_{d_j} has a high similarity to the text segments of l' from e_{d_i}.

(ii) Rank the elements from G'^d according to their relevance factor for d and add the elements with high relevance factor to g.

The relevance factor is defined by:

$$RF(e_{d_j}) = (1 - H(\theta_{d_i}, \theta_{d_j})) \cdot J(g_i, g_j) \cdot f(t(G^d)),$$

where $H(\theta_{d_i}, \theta_{d_j})$ is the Hellinger Distance, $J(g_i, g_j)$ the Jaccard index, and $f(t(G^d))$ the frequency of a relation between relations in the domain of two related documents d_i and d_j.

4 Case Study

This section presents a short case study of UTA linking each document to a specific DB and filling the database with entities and relations directly extractable from the document as well as associative data from related documents. The case study portrays an iterative DB construction. The goal is to identify entities and relations for a Wikipedia article without extracting entities and relations directly from the article itself.

uta is a Java program implementing the approach of UTA using the library MALLET [8] for topic modeling with the following parameters: (i) $\alpha = 0.01$, (ii) $\beta = 0.01$, (iii) topics $k = 100$, and (iv) number of iterations for the model in MALLET $= 1000$.

uta selects documents from $\mathcal{D}$ and adds them to D^d having a similarity in topic proportions to document d of $H(\theta_d, \theta_{d_j}) < t_1$, where t_1 is 0.8. The thresholds t_2 for the Jaccard Index $J(G, G_j)$ is set to 0.2.

4.1 Document Corpus Preparation

uta uses data from DBpedia [4] storing structured information as RDF triples and link them to each article in Wikipedia with a set of RDF triples. The triples from DBpedia serve as the ground truth.

The test corpus $\mathcal{D}$ contains 100 Wikipedia articles where each article has a link to the corresponding empty DB. The articles are about the automotive domain with a connection to the German car brand BMW. uta extracts the text of the Wikipedia articles and uses a parser to exclude HTML tags. Then, uta creates for each article an object containing the text of the article.

Having the corpus $\mathcal{D}$, containing all documents, we use MALLET to pre-process text documents by (i) lowercasing all characters, (ii) tokenizing the result, and (iii) eliminating tokens part of a stop-word list which contains 524 words. After pre-processing, uta randomly gets a document d from the corpus and generates a topic model for the remaining documents $\mathcal{D} \setminus \{d\}$. uta estimates the topic proportions for each document using MALLET. Afterwards, uta links the text with the topic proportions.

4.2 Iterative DB Construction

uta performs the following steps to iteratively fill the document-specific DB g of document d: (i) Estimate d's topic proportion θ_d, (ii) identify documents which are similar to d using the Hellinger distance and add them to D^d, (iii) identify similar documents by searching for isomorph subgraphs of g in $\mathcal{G} \setminus \{g\}$. After applying IE techniques to extract *extractable data*, uta extends the DB of each document in the iterative fashion by adding new data from other DBs. Associative data might be a suitable symbolic content description but we evaluate uta's performance against the annotations of a human annotator who often adds different data to a document's DB.

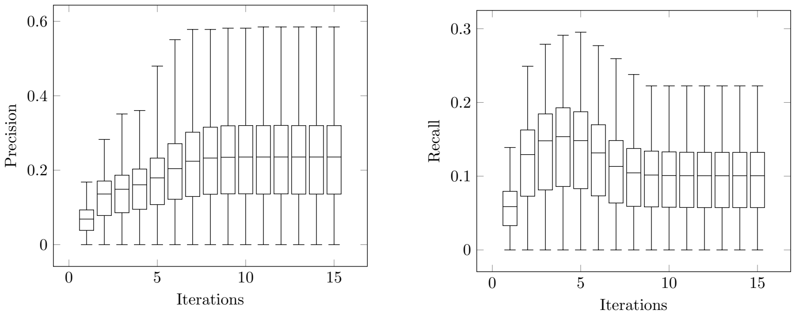

Fig. 1: Iterative KB construction of all 100 documents of the corpus $\mathcal{D}$. The left plot presents the precision of `uta`. Right plot presents the recall.

We perform for all 100 documents the iterative KB construction and present the precision and recall as Box plots in Figure 1. For most documents, the IE process was unable to extract all entities and relations. For some documents, the IE process was unable to extract any entity. Hence, the lower bound in the Box plots is 0 in Figure 1. For some documents, we have a precision of 0.6 and a recall of 0.29. In average, the precision of `uta` is 0.24 and the recall 0.11. `uta` reaches a fixed point at the latest after 15 iterations.

5 Conclusion

We present an unsupervised text annotation approach which links each document to a graph database containing entities and relations representing a content description. Each database represents data that is directly extractable from the document itself and data that is iteratively obtained from related documents. The performance of UTA depends on the number of topics, the two thresholds for the Hellinger distance and the Jaccard index, and the quality of information extraction techniques. UTA shows potential in being able to generate a symbolic content description in an unsupervised fashion for documents, but requires improvements to achieve higher precision and recall.

UTA can be used from knowledge graph systems to link documents with a corresponding symbolic content description containing extractable and associative data.

We are interested in extending UTA to an approach taking an arbitrary query as input and responding with a set of documents answering the query. This approach requires the linking of documents and the corresponding symbolic document description.

References

[1] 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.

[2] Ce Zhang. "DeepDive: a data management system for automatic knowledge base construction". PhD thesis. The University of Wisconsin-Madison, 2015.

[3] 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.

[4] Jens Lehmann and Robert Isele and Max Jakob and Anja Jentzsch and Dimitris Kontokostas and Pablo Mendes and Sebastian Hellmann and Mohamed Morsey and Patrick van Kleef and Sören Auer and Chris Bizer. "DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia". In: *Semantic Web Journal* (2014).

[5] 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, p. 3.

[6] 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.

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

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

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