Cross-Document Co-reference Resolution using Latent Features

Axel-Cyrille Ngonga Ngomo[♢], Michael Röder[♠♢], and Ricardo Usbeck[♠♢]

♢ University of Leipzig, Germany, ♠ R & D, Unister GmbH, Leipzig, Germany,
email: {ngonga|usbeck}@informatik.uni-leipzig.de

Abstract. Over the last years, entity detection approaches which combine named entity recognition and entity linking have been used to detect mentions of RDF resources from a given reference knowledge base in unstructured data. In this paper, we address the problem of assigning a single URI to named entities which stand for the same real-object across documents but are not yet available in the reference knowledge base. This task is known as cross-document co-reference resolution and has been addressed by manifold approaches in the past. We present a preliminary study of a novel take on the task based on the use of latent features derived from matrix factorizations combined with parameter-free graph clustering. We study the influence of different parameters (window size, rank, hardening) on our approach by comparing the F-measures we achieve on the N^3 benchmark. Our results suggest that using latent features leads to higher F-measures with an increase of up to 20.5% on datasets of the N^3 collection.

1 Introduction

The Document Web contains a large amount of information that is still not available on the Web of Data. For example, open extraction frameworks for unstructured data have been shown to harvest a considerable amount of new triples pertaining to real-objects for which no URI is available [3]. While no URI has been assigned to the said real-world objects, facts pertaining to these objects can be distributed across manifold data sources. Hence, simple URI generation approaches based on the labels of named entities can easily fail to generate the same URI when relying on two different labels that stand for the same real-world object. For example, simple URI generation schemes based on strings would fail to generate the same URI when presented with the strings "P. Diddy" and "Puff Daddy" as labels for resources. Moreover, they would generate the same URI for "Golf" across different documents even if the "Golf" stood for the sport in some documents and for the car in others. In literature, detecting that two labels stand for the same real-object even across documents is referred to as *cross-document co-reference resolution* (CDCR) [1,2]. While a large number of CDCR approaches have been developed in previous works (see Section 2), none of the current approaches makes use of latent features to detect whether two labels

stand for the same real-object. In previous work, latent features have yet been shown to be able to generate reliable representations of real-world objects [9].

In this paper, we address the aforementioned research gap by presenting the first CDCR approach based on latent features. Our approach represents entity mentions as bags of words. Each entity mention is then regarded as a vector in the space spanned by all words used to describe at least one entity mention. In the subsequent step, we compute the latent features of the entity mentions. The similarity of the latent representation of the entity mentions is then transformed into a similarity graph which is clustered by using BorderFlow [8], a parameter-free graph clustering approach. All entity mentions which belong to the same cluster are regarded as mentions of the same real-world object and are assigned to the same URI. Our approach is open-source and available at `http://github.com/AKSW/CoreferenceResolution`.

The rest of this paper is organized as follows: First, we give an overview of previous CDCR approaches. Then, we present our approach in detail. In Section 4, we evaluate our approach on the N^3 benchmark dataset [13] and compare it with a baseline approach. We conclude the paper and discuss future work in Section 5.

2 Related Work

In the following section, we will provide an overview over recent approaches towards CDCR with a focus on their underlying techniques w.r.t. the semantic and syntactic features they exploit.

Mayfield et al.'s [6] CDCR approach comprises five stages: (1) intra-document processing, i.e., identification of mentions of entities, (2) entity pairs filtering, i.e., discarding of possible entity mappings to reduce computational costs, (3) calculating features of entities, (4) classification of entity matching by machine learning techniques and (5) clustering of entities to map each mention to the same equivalence class. Unfortunately, the authors evaluated their approach in the ACE 2008 English named entity recognition task which is no longer available. There, the approach achieved a value metric of 54.8 [10].

Haghighi et al. [4] present an unsupervised approach based upon a generative process which is capable to use modular syntactic and semantic features making use of latent information. For every document, the generative process creates a number of entities mentioned in the text. For every mention a noun phrase is created. However, since the inference algorithm only uses these noun phrases, their approach lacks on taking a larger context into account.

Rahman et al. [12] introduce an approach which incorporates *world knowledge* into two baseline CDCR algorithms. Thereby, the authors use YAGO[1] and FrameNet[2] as underlying knowledge bases. Afterwards, they use a mention-entity

[1] `http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/`

[2] `https://framenet.icsi.berkeley.edu/fndrupal/`

pair classifier and a cluster-ranking model. The results show an improvement over each baseline.

Singh et al. [15] present an approach consisting of (1) a large scale distributed inference mechanism based on Markov chain Monte Carlo methods and (2) they introduce sub-entity and super-entity variables representing clusters which are used to distribute or collect certain entities on a specific part of the machine cloud. Furthermore, they evaluate their approach on a 1.5 million document comprising web crawl using anker tags to Wikipedia as gold standard. Nevertheless, the authors approach misses the opportunity to consider latent features resulting in large computational costs w.r.t. the size of the resulting Markov chain.

Lee et al. [5] present an approach not only capable of co-referencing entities but also events. Their idea is based upon linear regression which is used to merge clusters of entities. Furthermore, the authors featurize entities via semantic role labeling. Their approach is able to co-reference entities intra- and inter-document-wise. Although the authors claim to be better than the state-of-the-art with respect to the CoNLL 2011 shared task [11] their published corpus is not available anymore.

In 2013, Beheshti et al. [2] provide a systematic analysis of state-of-the-art CDCR systems. The survey provides an in-depth structurization of the underlying methods and algorithms, which are widely used to solve CDCR problems on large scale. Furthermore, the authors highlight certain Big Data challenges, e.g., large amounts of pair-wise string similarity calculations and costly classification algorithms.

Normally, these approaches are based on a trained set of parameters for semantic and syntactic similarity algorithms. Recently, Andrews et al. [1] describe an approach towards CDCR, here called entity clustering, that relies on learning parameters from test data without the need for training data. The generative process within assumes a mutation of semantic context and syntactic similarity while generating the documents with cross-referenced entities. Afterwards, the authors deploy a block Gibbs sampler to infer the clusters. Unfortunately, this approach is only empirically evaluated.

With respect to the clustering aspect of this paper, Schaeffer [14] provides an exhaustive overview of common graph-clustering algorithms and their use cases.

To the best of our knowledge, we present the first paper on CDCR based on latent features, matrix decomposition as well as graph-clustering.

3 Approach

In this section, we present our approach to CDCR in more detail. We introduce the notation necessary to understand the approach as required by each section. Figure 1 gives an overview of the five steps that underly our approach. In a first step, a Matrix M is generated containing the context of every entity mention. After that, this matrix is decomposed into two smaller matrices L and R with $M \approx LR^\top$. In parallel, a second matrix S is created which contains the pairwise

similarities of the labels of the entity mentions. These matrices are used to generate a symmetric graph G in which (1) every entity mention is a node and (2) two nodes are connected if their similarity is higher than a certain threshold. G is finally clustered. Mentions that belong to the same cluster are considered to be mentions of the same entity. Hence, they are all assigned the same URI.

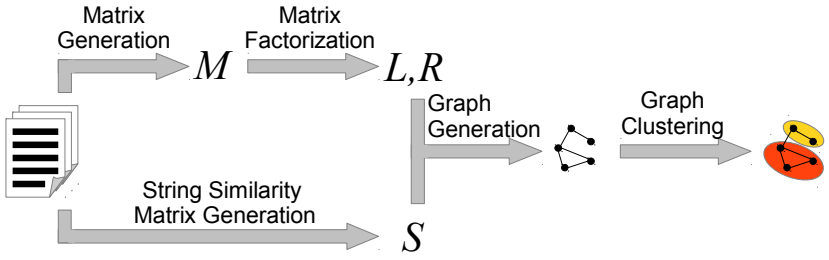


Fig. 1: The five steps of our approach.

3.1 Matrix Generation

The first step of our approach consists of generating a matrix which describes the context of every named entity mention inside the texts by means of a bag of words. To this end, the given corpus is preprocessed by tokenizing the documents, removing stop words and indexing the remaining tokens. In these tokenized documents, the context of a named entity mention is defined as the multiset of tokens inside a window with the size $\pm\sigma$ that is centered on the named entity's tokens. The contexts are stored in a matrix M containing a row for every named entity mention and a column for every indexed word. The entries of the matrix are the counts of the words inside the entity mention's context. As an example, let us consider the sentence

Example 1. `Yesterday, VW's CEO presented the new Golf in Munich.`

from which the stopwords {`the`, `in`} are removed. For the window size $\sigma = 1$, we get the bag-of-word multiset {`new` (1), `Munich` (1)} as representation of "Golf". Within the vector space spawned by (`presented`, `new`, `Munich`, `Germany`), this mention has the vector representation $(0, 1, 1, 0)$. In the following, we will consider five entity mentions g_1, g_2, g_3, g_4 and g_5 labelled with the same word "golf" as example. These entity mentions will be assumed to be represented by the vectors $g_1 = (2, 2, 2, 0)$, $g_2 = (1, 0, 0, 1)$, $g_3 = (0, 0, 0, 1)$, $g_4 = (1, 0, 0, 0)$ and $g_5 = (0, 1, 1, 0)$.

3.2 Matrix Factorization

The matrix M is now a matrix of dimensions $n \times m$ (denoted $M(n, m)$). The goal of a matrix factorization is to compute the matrices $L(n, \rho)$ and $R(m, \rho)$

such that $M \approx LR^\top$. We call $\rho \in \mathbb{N}\backslash 0$ the *rank* of the factorization. Several approaches have been used to factorize matrices. Here, we loosely follow the tensor factorization approach presented in [9]: Given two matrices L and R that are supposed to be the factors of M, the overall quadratic error of the approximation is the square Frobenius norm of $E = M - RL^\top$, i.e., $||E||_F^2 = ||M - RL^\top||_F^2$. Previous works have shown that to prevent overfitting, the error function to minimize must be extended. While several approaches have been suggested to this end, we adopt the error expression given by $||E||_F^2 - \frac{\lambda}{2}(||R||_F^2 + ||L||_F^2)$, where $\lambda \in [0, 1]$ controls how well L and R fit M. Thus, the error derivatives are as follows:

$$\frac{\partial e_{ij}}{\partial r_{ik}} = -2e_{ij}l_{jk} + \lambda r_{ik} \tag{1}$$

and

$$\frac{\partial e_{ij}}{\partial l_{jk}} = -2e_{ij}r_{ik} + \lambda l_{jk}. \tag{2}$$

We can now adopt a gradient descent approach to update the matrices L and R and reduce the error they lead to by overwriting each l_{ik} resp r_{jk} as follows:

$$l_{jk} \leftarrow l_{jk} - \alpha\frac{\partial e_{ij}}{\partial l_{jk}} = l_{jk} + \alpha\left(2\sum_{i=1}^{n} e_{ij}r_{ik} - \lambda l_{jk}\right) \tag{3}$$

and

$$r_{ik} \leftarrow r_{ik} - \alpha\frac{\partial e_{ij}}{\partial r_{ik}} = r_{ik} + \alpha\left(2\sum_{j=1}^{j} e_{ij}l_{jk} - \lambda r_{ik}\right). \tag{4}$$

We initialize L and R with random entries between 0 and $\max m_{ij}$. For our example, we get

$$M = \begin{pmatrix} 2\,2\,2\,0 \\ 1\,0\,0\,1 \\ 0\,0\,0\,1 \\ 1\,0\,0\,0 \\ 0\,1\,1\,0 \end{pmatrix}. \tag{5}$$

For $\rho = 2$, our approach computes

$$L = \begin{pmatrix} 1.385 & 1.102 \\ -0.006 & 0.501 \\ 0.079 & -0.051 \\ -0.234 & 0.712 \\ 0.933 & -0.168 \end{pmatrix} \quad \text{and} \quad R = \begin{pmatrix} 0.331\ 1.406 \\ 1.059\ 0.446 \\ 1.118\ 0.363 \\ 0.062\ 0.066 \end{pmatrix}. \tag{6}$$

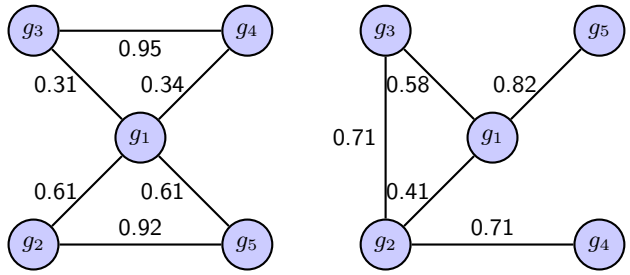
The intuition behind our approach is that L is a better and compressed description of the entity mentions than M. Hence, we now use L in combination with a string similarity function to compute the similarity of entity mentions.

3.3 String Similarity Matrix

The string similarity matrix S is an optional feature of our approach. Each entry s_{ij} of S describes the similarity between the label of the ith and the jth entity in our input corpus. Assuming a symmetric string similarity function such as the 3-gram similarity (which we use in our experiments), we also get a symmetric string similarity matrix S. We assume $s_{ij} = 1$ if no string similarity is specified. $s_{ij} = 1$ also holds for our example, as all mentions are labelled with "golf".

3.4 Graph Generation

The aim of the graph generation is to generate a similarity graph $G = (V, E, w)$ that will allow detecting mentions of the same real-world object through clustering. The set of vertices of V is the set of entity mentions in our corpus. We define the weight function $w : V \times V \to [0, 1]$ as $w(v_i, v_j) = s_{ij} \times \frac{l_{(i,\cdot)} \cdot l_{(j,\cdot)}}{||l_{(i,\cdot)}|| \times ||l_{(j,\cdot)}||}$, where $l_{(i,\cdot)}$ is the ith row-vector of L and stands for the latent description of the ith entity mention in the corpus. Given that many graph clustering approaches are polynomial in the number of edges, we can control $|E|$ by only setting an edge between v_i and v_j if $w(v_i, v_j) \geq \theta \in [0, 1]$. For $\theta = 0.3$ and $\rho = 2$ we end up with the graph displayed in Figure 2(a). As comparison, Figure 2(b) shows the graph obtained with by setting $L = M$, i.e., generating G without using latent features.



(a) Graph generated using $\rho = 2$

(b) Graph generated using M instead of L

Fig. 2: Graphs generated by our approach for the example dataset.

3.5 Graph Clustering

We now cluster the graph G to detect mentions that stand for the same real-world object. Our approach can rely on any graph clustering approach. In our current implementation, we rely on the BorderFlow algorithm [8] because it is parameter-free. BorderFlow regards any set $C \subseteq V$ as having a *border* $b(C) =$

$\{v \in C : \exists u \in V \backslash C \text{ with } (v,u) \in E\}$. The *flow* $\Omega(C_1, C_2)$ between two sets $C_1 \subseteq V$ and $C_2 \subseteq V$ is defined as $\Omega(C_1, C_2) = \sum_{v \in C_1, u \in C_2} w(v,u)$. Based on these definitions, BorderFlow implements a local graph clustering paradigm by mapping each node $v \in V$ to the set of nodes $C \subseteq V$ that is such that $v \in C$ and C is a node-maximal set w.r.t. the function

$$bf(C) = \frac{\Omega(b(C), C)}{\Omega(b(C), V \backslash C)}. \tag{7}$$

While finding the optimal C for each v can be very time-consuming, the heuristic presented in [7] allows determining an approximation of C in an efficient manner. We employ this heuristic herein.

Now, the result of BorderFlow is not a partitioning of the graph. Rather, clusters may overlap. We thus employ a *hardening* approach to generate a partitioning of the input graph. To this end, each node $v \in V$ which belongs to two different clusters C_1 and C_2 is assigned to C_1 iff

$$bf(C_1 \cup \{v\}) + bf(C_2 \backslash \{v\}) \geq bf(C_2 \cup \{v\}) + bf(C_1 \backslash \{v\}). \tag{8}$$

In all other cases, v is assigned to C_2. We call this form of hardening *flow maximization*. Other forms of hardening can be conceived of, e.g., minimizing the number of unions operations that need to be carried out to achieve a partitioning of the graph (*set-based*).

For our example, we get the clusters $\{g_1, g_5\}$ and $\{g_2, g_3, g_4\}$ for $\rho = 2$ when using BorderFlow with any partitioning approach. If we replace L with M, we get the clusters $\{g_1\}$, $\{g_2, g_4\}$ and $\{g_3, g_5\}$. This result on toy data already suggests that matrix factorization leads to results that differ from those gathered when using raw data. In the subsequent section, we show empirically that using L to generate G leads to more accurate results than using M to generate G.

4 Evaluation

4.1 Experimental Setup

Goals The goal of our experiments was two-fold. First, we wanted to measure the effect of the different parameters on our approach. Moreover, we wanted to know whether the factorization outperforms a comparable baseline. To achieve the first goal of our experiments, we conducted experiments where we varied the rank ρ as well as the window size σ while keeping all other parameters fixed. We addressed the second goal by creating a baseline as follows: We ran our pipeline as described in the sections above with the sole difference that (1) we did not carry out a factorization and (2) we use M instead of L as input for the graph clustering. All other steps (matrix generation, graph generation, graph clustering) remained unchanged. The similarity threshold for the graph generation is set to $\theta = 0.1$ for all our experiments.

Datasets We use the three corpora of the N^3 collection [13] in our experiments.

- The **News-100** corpus comprises 100 German news articles from `news.de`. Each of these articles contains the German word "Golf"—a homonym that has three different meanings inside these documents. The word could mean (a) a gulf, e.g., the Mexican gulf, (b) the ball sport or (c) a compact car of the German manufacturer Volkswagen. This is clearly the most difficult dataset, as many resources share exactly the same name but have different meanings.
- The **Reuters-128** corpus contains 128 English economy news articles from the Reuters news agency. The documents in this dataset are smaller than the ones from the News-100 corpus providing a shallow context.
- The third corpus, **RSS-500**, contains 500 documents each with only one sentence. The sentences were randomly chosen from a larger amount of RSS news feeds, as described in [3]. Every sentence contains exactly two named entities.

Table 1 provides further detailed information about the corpora. On average, each named entity occurs nearly 5 times in the News-100 corpus. Within the Reuters-128 corpus nearly two mentions per named entity exist on average while in the RSS-500 corpus only every tenth entity is mentioned more than once.

Table 1: Features of the corpora

	News-100	**Reuters-128**	**RSS-500**
Documents	100	128	500
Tokens	48199	33413	31640
Entities	362	444	849
Mentions	1655	880	1000

4.2 Results

Influence of rank In our first series of experiments, we fixed the window size to 4 and measured the influence of the rank ρ on the precision, recall and F-measure. The left side of Figure 3 shows the results of our experiments on the three datasets. Most importantly, our results show that we outperform the baseline in most settings. We achieve the best increase of performance on the RSS-500 corpus, where we achieve a 20.5% increase in F-measure over the baseline. This result suggest that our approach does not tend to overgeneralize through the compression on information that is carried out during the factorization. Instead, our results suggest that we get rid of a significant amount of noise while factorizing. Our results on the other two datasets show that we also achieve a better F-measure (increases of 18.2% on Reuters-128 and 6.3% on News-100, see Table 2). An analysis of the results reveals that this increase is mostly due to the

significant increase in precision that we achieve in most settings. On the other hand, our recall is rarely ever worse than that of the baseline. This suggests that BorderFlow tends to generate smaller clusters with factorization than when the baseline approach is used. We measure the statistical significance of our results using a Wilcoxon signed rank-test with 95% confidence. Our results are significant in all cases.

Influence of window size In this experiment, we set the rank to 100 for all experiments and measured the effect of the window size on the overall F-measure of our approach. The right half of Figure 3 shows the results of this series of experiments on the three datasets. Overall, our results suggest that for this rank, the window size does not have a major influence on the F-measure. This also seems to hold for other ranks. Interestingly, a small window size seems to lead to good results in most cases when we use the factorization. While we assume that this might be due to the factorization being able to convert the two words within the window to their latent This result indicates that small window sizes suffice for our approach to achieve better F-measures than the baseline on the CDCR problem. This might mean that a small set of words is already sufficient to disambiguate resources across different documents.

4.3 Effect of hardening

In all results presented above, we used a hardening based on the borderflow ratio. We also implemented the set-based hardening mentioned above and compared the results we achieve with this hardening. Overall, our results suggest that the borderflow-maximization approach that we used for hardening generates the best results both for the baseline and our approach. Moreover, we outperform the baseline independently from the hardening used.

Table 2: Best improvements in F-measure of our approach (OA) over the baseline (BL)

	Flow Maximization		**Set-Based**	
	BL	OA	BL	OA
News-100	25.86	**32.21**	23.87	**28.81**
Reuters-128	47.89	**66.16**	47.00	**56.65**
RSS-500	71.11	**91.62**	69.57	**85.71**

Discussion Overall, our initial results suggest that we indeed outperform the proposed baseline by using matrix factorization (see Table 2). Still, many questions do remain open. The most important question that we did not address is

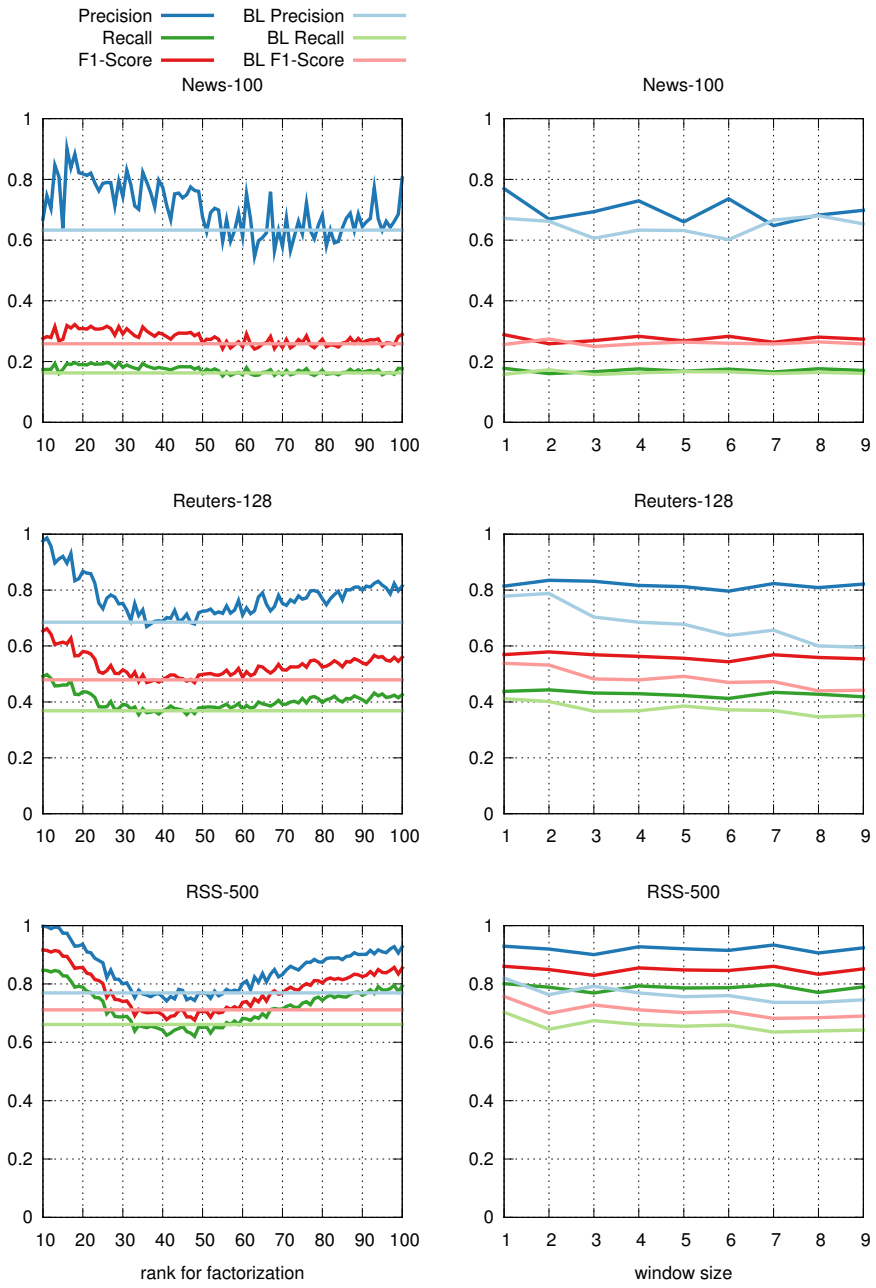


Fig. 3: Precision, recall and F1-score of our approach with different ranks (left) and different window sizes (right) compared to the baseline (BL). The diagrams show the results for the flow maximization hardening.

when should a high rank be used? First, in our experiments, $\rho = 10$ was sufficient across all datasets to outperform the baseline. To the best of our knowledge, finding the optimal rank for a factorization problem is an open question. Nevertheless, we think that the answer to this question lies in the amount of information contained in the corpus. The higher the information density of a corpus, the higher the rank required to characterize entity adequately. A second question that remains unanswered is whether we can improve the results of the factorization by considering known resources in the dataset. We will address this question in future work by disambiguating using a combination of textual information and Linked Data.

5 Conclusion

In this paper, we presented a CDCR approach based on latent features. We showed that our approach can outperform our baseline by more than 10% F-measure. We will use our approach to complement the entity linking framework [16] when it is used in batch mode, i.e., over a document corpus at once. Moreover, we will develop means to detect an appropriate rank for factorization. To this end, we plan to use the derivative of the mean squared error $||M - LR^\top||_F^2$. Finally, we will develop a deterministic approach to initialize L and R. Preliminary results on random matrices show that we can already reduce the initial value of $||E||_F^2$ by more approximately 40%, leading to a significantly faster convergence of the factorization.

Acknowledgments

This work has been supported by the ESF and the Free State of Saxony and the FP7 project GeoKnow (GA No. 318159).

References

1. N. Andrews, J. Eisner, and M. Dredze. Robust entity clustering via phylogenetic inference. In *Association for Computational Linguistics (ACL)*, 2014.
2. S.-M.-R. Beheshti, S. Venugopal, S. H. Ryu, B. Benatallah, and W. Wang. Big data and cross-document coreference resolution: Current state and future opportunities. *CoRR*, abs/1311.3987, 2013.
3. D. Gerber, A.-C. Ngonga Ngomo, S. Hellmann, T. Soru, L. Bühmann, and R. Usbeck. Real-time rdf extraction from unstructured data streams. In *ISWC*, 2013.
4. A. Haghighi and D. Klein. Coreference resolution in a modular, entity-centered model. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 385–393. Association for Computational Linguistics, June 2010.
5. H. Lee, M. Recasens, A. Chang, M. Surdeanu, and D. Jurafsky. Joint entity and event coreference resolution across documents. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 489–500, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

6. J. Mayfield, D. Alexander, B. J. Dorr, J. Eisner, T. Elsayed, T. Finin, C. Fink, M. Freedman, N. Garera, P. McNamee, et al. Cross-document coreference resolution: A key technology for learning by reading. In *AAAI Spring Symposium: Learning by Reading and Learning to Read*, pages 65–70, 2009.
7. A.-C. Ngonga Ngomo. Parameter-free clustering of protein-protein interaction graphs. In *Proceedings of Symposium on Machine Learning in Systems Biology 2010*, 2010.
8. A.-C. Ngonga Ngomo and F. Schumacher. Borderflow: A local graph clustering algorithm for natural language processing. In *CICLing*, pages 547–558, 2009.
9. M. Nickel, V. Tresp, and H.-P. Kriegel. Factorizing yago: scalable machine learning for linked data. In *WWW*, pages 271–280, 2012.
10. NIST. Automatic Content Extraction 2008 Evaluation. http://www.itl.nist.gov/iad/mig//tests/ace/.
11. S. Pradhan, L. Ramshaw, M. Marcus, M. Palmer, R. Weischedel, and N. Xue. Conll-2011 shared task: Modeling unrestricted coreference in ontonotes. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning (CoNLL 2011)*, Portland, Oregon, June 2011.
12. A. Rahman and V. Ng. Coreference resolution with world knowledge. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 814–824, 2011.
13. M. Röder, R. Usbeck, S. Hellmann, D. Gerber, and A. Both. N^3 - a collection of datasets for named entity recognition and disambiguation in the nlp interchange format. In *The 9th edition of the Language Resources and EvaluationConference, 26-31 May, Reykjavik, Iceland*, 2014.
14. S. E. Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27–64, 2007.
15. S. Singh, A. Subramanya, F. Pereira, and A. McCallum. Large-scale cross-document coreference using distributed inference and hierarchical models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 793–803, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.
16. R. Usbeck, A.-C. Ngonga Ngomo, S. Auer, D. Gerber, and A. Both. AGDISTIS - Agnostic Disambiguation of Named Entities Using Linked Open Data. In *International Semantic Web Conference*. 2014.