Document clustering as a record linkage problem

Nikiforos Pittaras, George Giannakopoulos, Leonidas Tsekouras
Institute of Informatics and Telecommunications
N.C.S.R. Demokritos
Greece
{pittarasnikif,ggianna,ltsekouras}@iit.demokritos.gr

Iraklis Varlamis
Department of Informatics and Telematics
Harokopio University of Athens
Greece
varlamis@hua.gr

ABSTRACT

This work examines document clustering as a record linkage problem, focusing on named-entities and frequent terms, using several vector and graph-based document representation methods and k-means clustering with different similarity measures. The JedAI Record Linkage toolkit is employed for most of the record linkage pipeline tasks (i.e. preprocessing, scalable feature representation, blocking and clustering) and the OpenCalais platform for entity extraction. The resulting clusters are evaluated with multiple clustering quality metrics. The experiments show very good clustering results and significant speedups in the clustering process, which indicates the suitability of both the record linkage formulation and the JedAI toolkit for improving the scalability for large-scale document clustering tasks.

CCS Concepts

•Information systems → Clustering; Document representation;

Keywords

Clustering; Record Linkage; Entity Resolution;

1. INTRODUCTION

Document clustering aims at grouping a set of documents into coherent groups (called clusters), so that documents in the same group are more similar to each other than to those in other groups. As a consequence, clustering algorithms rely on a measure of similarity or distance between the examined documents, or on an adjacency/connectivity matrix, which conveys information that two documents are somehow connected (related) or not. In a number of research and real-life text analysis scenarios, clustering can be applied to group documents e.g. based on their common topic [Kuang et al., 2015], underlying event [Daniel et al., 2003] or other criterion. Different text comparison approaches can heavily differentiate the result of clustering over the same document collection, making the task of understanding whether two texts are talking about the same topic or are somehow related, a critical yet challenging step for the clustering process.

Record linkage (also termed entity resolution) is the task of searching across different data sources (e.g., data files, documents and databases) and locating the records that refer to the same entity. It is a step towards data integration, disambiguation and correction and can be based on rules (deterministic), fuzzy matching (probabilistic) or machine learning [Brizan and Tansel, 2006]. The intuition behind treating document clustering as a record linkage problem is the following: considering that the entity described in a document is a topic/event, the task of clustering documents can be regarded as a task of linking information (records) over those topics/events. This allows to employ blocking and refinement techniques from Record linkage research [Papadakis et al., 2016] to perform maximally efficient clustering. This work: (i) formulates text clustering as a record linkage problem, (ii) explains the analogy between the two problems and (iii) demonstrates how record linkage can contribute to document clustering over different settings.

2. RELATED WORK

A representation model and a similarity measure are the prerequisites for **text document clustering**. The Vector Space Model (VSM) and unigram (Bag-of-Words *BoW*) or n-gram (multi-word or multi-character) models have been widely used in the related literature [Gomaa and Fahmy, 2013]. Semantic relatedness measures for text that use linguistic resources and ontologies also capture word dependence [Tsatsaronis et al., 2010] and word embeddings that create high-dimensional representations of words aim to capture word relationships and linguistic regularities [Kusner et al., 2015]. Cosine similarity is usually employed to calculate the similarity of vector representations.

N-gram graphs (ngg) [Giannakopoulos and Karkaletsis, 2009] is a graph-based alternative to vector representation for text, which captures the word order by connecting neighboring n-grams with edges. [Schenker et al., 2005] performed text clustering and classification tasks using graph representation models and graph edit distance metrics. [Giannakopoulos and Karkaletsis, 2009] represented texts as n-gram graphs, using a sliding window of length n and compared their graphs using metrics such as Value Similarity, Normal Value Similarity, Value Ratio and Size Similarity. In [Tsekouras et al., 2017], the authors used only the most informative terms in order to reduce the n-gram graph com-

DocEng '18, August 28–31, 2018, Halifax, NS, Canada

ACM ISBN 978-1-4503-5769-2/18/08…$15.00

DOI: https://doi.org/10.1145/3209280.3229109

plexity, without losing significant information.

A lot of works on small text clustering in Twitter associate document clusters with events (e.g. [Becker et al., 2011]). [Reuter et al., 2011] formulated the event identification problem as a **record linkage** task in short texts, employed a time-based blocking strategy to reduce the number of pairs of documents that are compared, and introduced a composite similarity measure that accounts textual, time and location features. Record linkage systems perform many pairwise similarity computations but also use clustering and schema mapping algorithms. Record linkage has also been employed by [Bi et al., 2016] for the classification of XML Feeds and event detection.

3. PROPOSED METHOD

The record linkage formulation is applied to the document clustering problem, as follows:

DEFINITION 1. *Given a set E of (unknown) entities, a (known) set D of (unstructured) documents (vs. entity descriptions), and an unknown (but a posteriori evaluable) mapping function $f : D \rightarrow E$, which maps every description to one entity, we define a record linkage problem as the search of a partitioning of $D = D_0 \cap D_1 \cap ... \cap D_n$, where each $d_i, d_j \in D_i, f(d_i) = f(d_j)$.*

In the above formulation: (i) the number (or nature) of entities is unknown before solving the problem, which is similar to clustering documents in previously unknown groups, (ii) the documents are unstructured/schema-less, whereas records are usually structured, (iii) only posterior evaluation is possible, since we can only examine whether a set of documents maps to the same entity after the clustering, (iv) "entity" can be equivalent to the document's "topic" or "event", which means that a document mapped to a specific topic/event is expected to cluster together with all other documents mapped to the same topic/event.

This work builds on the JedAI toolkit [Papadakis et al., 2017], an end-to-end Entity Resolution (ER) toolkit for record linkage. JedAI combines a multitude of data source representation models, similarity metrics and clustering techniques in a modular fashion, aiming to provide an out-of-the-box ER solution. The JedAI execution pipeline in this case, consists of the following steps: (i) Data Reading: input text documents are loaded as data instances. (ii) Block Building: generation and partitioning of comparisons between input texts, as overlapping blocks). (iii) Block Cleaning and Comparisons Cleaning: the "Standard Blocking" method of JedAI coupled with Weighted Edge Pruning [Papadakis et al., 2016] allow to discard redundancies generated during block building and reduce the total number of comparisons, (iv) Entity Matching: entities in the cleaned blocks are represented and compared in a pair-wise manner per block. The "Profile Matching" method along with an entity/document similarity metric is used to create a document similarity graph as an input for clustering. (v) Entity-Based Document Clustering: This step uses the similarity graph and the Ricochet Sequential Rippling algorithm [Wijaya and Bressan, 2009] to group documents into equivalence clusters.

To understand the gains and losses of the proposed approach we conduct a number of experiments, expected to answer two main questions: (i) Does the application of record linkage blocking and refinement methods improve the efficiency/speed of the clustering process? (ii) How does it affect the effectiveness/accuracy of the clustering in identifying documents that map to the same topic/event/etc?

A baseline for comparison, is the Vector Space Model (VSM) representation with and without TF-IDF weighting, considering both Bag-of-Words (BoW) and Bag of n-grams (considering token trigrams, $n \in \{1, 3\}$) and vectors are compared using the cosine distance metric.

Graph-based similarity is evaluated on word and character 3-gram graph representations [Giannakopoulos, 2009], and graphs are compared using the Normalized Value Similarity (NVS) measure [Giannakopoulos and Karkaletsis, 2009]. NVS compares two graph models by considering the number common edges between them along with the weights of the common edges (Value Similarity), while discounting the potential size difference between the graphs (i.e., normalized by the Size Similarity). Both graph and vector representations are applied on the raw textual document contents as well as on the extracted named-entity data.

Finally, Mixed N-gram Graphs, which combine in a single N-gram graph the named-entities extracted from the document and the top (TF-IDF-weighted) terms in the raw text. All other terms are replaced with a placeholder string and the resulting word unigram graph that is constructed contains nodes that correspond to top terms from the raw text, named entities and a single node for all non-important content (placeholder string). Graphs (i.e. documents) are compared using NVS measure.

4. EXPERIMENTAL SETUP

Experiments have been performed on: (i) the 20 Newsgroups[1] (20N) dataset sorted by date, which consists of about 20,000 news documents in English that span 20 broad topics (classes), and (ii) the MultiLing 2015[2] (ML15) multi-document summarization dataset that consists of 15 events from WikiNews described by documents in various languages. Only the test data from the 20N dataset have been employed, which comprises 7526 documents. From the second dataset only documents written in English, Spanish and French (the languages supported by OpenCalais) are kept, resulting to 400 documents that represent all the 15 events.

Two document preprocessing approaches are employed: (i) raw text and (ii) named-entities. JedAI handles records as name-value pairs, the name "content" has been matched with the document text as value, creating a single name-value pair for each input document. Alternatively, named-entities are extracted from documents, using the OpenCalais API[3], which returns among others, a `name` and a `type` attribute for each resolved entity and information on the entity's position in the source text. In order to prevent multi-word entities from being split up in a word representation model, the whitespace between words in the entity name is replaced by underscore. For example, the text "Elon Musk" maps to the name-value set {"name": "Elon_Musk", "type": "Person"}. Each document fed to JedAI contains many such sets, which is the standard representation type for many data record storage schemes.

[1] http://qwone.com/~jason/20Newsgroups

[2] http://multiling.iit.demokritos.gr/pages/view/1516/multiling-2015

[3] http://www.opencalais.com/

As a result, each cluster contains documents, which in turn are linked to entities (topics/events). Each cluster is mapped to the topic/event that is the majority of the cluster and if there is no majority in the cluster, the cluster is omitted. Clusters assigned to the same predicted topic/event are consequently merged, discarding duplicate documents.

In the evaluation phase, the resulting clusters along with their member documents are compared with the ground truth of each dataset (classes), and the method performance is measured using micro-averaged F-score, Clustering Precision (CPr - the number of pairs of documents that share a ground truth class) and Penalized Clustering Precision ($PCPr$) metrics [Hassanzadeh et al., 2009]. Merging clusters as described above does not result into favorable $PCPr$ scores, compared to the scores we get without merging. However, merging was necessary for the computation of the F-score.

5. EXPERIMENTAL RESULTS

Several experiments are performed in order to address the questions mentioned in section 3. Firstly, we fine-tune the similarity threshold that indicates whether two entities should be connected or not, to the value of 0.1, using only the smaller (but multilingual) ML15 dataset. Three sets of experiments on ML15 followed, using variable block refinement levels to measure the effect on the speed and effectiveness of the clustering. The first used no refinement mechanisms in the pipeline, while the other two adopted the size-based block purging procedure. From the two block refinement-based runs, the "standard" one uses a pf value of 0.005, which is the value used in every experiment with JedAI in this study. The other refinement run, named "half", uses a $\text{pf}' = 100 \times \text{pf} = 0.5$. For each experiment, the elapsed time, number of comparisons and the micro-average F-score (averaged across all configurations) are reported.

In order to assess the effectiveness of both the record linkage adaptation to the document clustering problem and the suitability of the JedAI framework for the task, experiments on both datasets have been performed. All the combinations of text representation methods and similarities have been evaluated using F-measure, CPr and $PCPr$ metrics and the ground truth classes of each dataset.

Table 1 illustrates the results on the time comparison experiments on the ML15 dataset. The "standard" block refinement configuration introduces a 97.4% reduction to the number of comparisons compared to not using block refinement. It also decreases the running time by 97.6%, with a 24.4% F-score performance reduction. The "half" refinement configuration prunes less comparisons than the "standard" one and improves the running time and number of comparisons by 63.4% and 55.19% respectively, at a performance cost of just 5.6%. Results show that the blocking-based refinement can be tuned to suit the needs for a variety of clustering tasks, with respect to balancing the scalability with the clustering quality, making the method attractive for large workloads as well as real-time applications.

Table 2 summarizes the performance of each representation model on both datasets. In the "method" column, "bow-n" entries indicate BoW unigram models for $n = 1$ (i.e. standard BoW) and BoW trigrams for $n = 3$. Entries containing ngg-xn denote ngg models using word ($x = \text{w}$) or character ($x = \text{c}$) tokens, in a unigram or trigram configuration as per the n value. The suffix denotes the data type used

block refinement	comparisons	time	mean F
standard	28,264	7.4	0.680
half	495,518	113	0.849
none	1,105,903	309	0.900

Table 1: Time performance (seconds), number of comparisons and micro-average F-score per block refinement configuration used, on the MultiLing dataset. All values are the mean of all representation model / input type combination.

by each configuration, i.e. "-t" for raw document text, "-e" for named-entities or "-et" for both text and named-entities. Each configuration is evaluated with respect to three evaluation metrics: micro averaged F-score (denoted with F in Table 2), CPr and $PCPr$. Bag of Words (BoW) and n-gram graph (ngg) configurations use the cosine and NVS similarity measures, respectively.

Mean F-score results are better on the 20N dataset than in ML15 across all configurations. However, better average CPr and $PCPr$ are achieved in ML15. This is mainly due to the differences in size and number of ground truth clusters in each datasets (400 vs 7526 documents and 15 vs 20 events/topics, in ML15 and 20N, respectively).

In ML15, using only named-entities consistently and considerably outperforms raw text models concerning F-score. This can be attributed to both the structured data provided by the extracted entities, as well as the relatively few data per event/topic in ML15. For all VSM models (BoW and TF-IDF), the unigram scheme outperforms trigram combinations of the input. Character ngg models consistently outperform word-based $nggs$, the latter of which, surprisingly, do not manage to capture the structure of their input, especially for the raw text case. The mixed input ngg performs well, but the entity unigram TF-IDF model is the best F-score-related performer, followed by its raw text variant. The approach managed to handle the multi-lingual aspect of the data remarkably well, achieving an F-score 0.974. Regarding CPr scores, the poorest performing configurations with respect to F-score produce the best CPr scores (i.e. tfidf3-t and ngg-w3-t). These configurations result in a small number of clusters in our experiments and this is the main driver for the high precision performance of the configuration, since few clusters translates to a high chance of placing documents that share the same cluster — in the ground truth — together in the output. The limited number of clusters however, corresponds to a low recall and F-score. This effect is also reflected by $PCPr$, which penalizes such cases severely. Discounting these perfect unit-scoring CPr configurations, the best configuration is the TFIDF named-entity unigram model, across all evaluation metrics.

With respect to the 20N dataset, and F-score performance, using only named-entity information is comparable to working on raw text. The best performing model is the TF-IDF text unigram model, followed by the mixed unigram ngg and the entity-based TF-IDF unigram models. Word ngg models outperform character-based counterparts for both text and named-entity input. TF-IDF unigram models outperform their trigram versions, while the opposite occurs for the BoW models. The worst performing configurations in ML15 (i.e. tfidf3-t and ngg-w3-t) fare better here, matching the average performance of all configurations, with tfidf3-t even achieving the best CPr and $PCPr$ score.

	MultiLing 2015			20 Newsgroups		
method	F	CPr	$PCPr$	F	CPr	$PCPr$
bow1-e	0.946	0.903	0.903	0.773	0.624	0.624
bow1-t	0.818	0.768	0.768	0.697	0.553	0.553
bow3-e	0.698	0.838	0.838	0.784	0.675	0.675
bow3-t	0.451	0.946	0.820	0.782	0.915	0.915
tfidf1-e	**0.974**	0.949	**0.949**	0.798	0.667	0.667
tfidf1-t	0.958	0.932	0.932	**0.834**	0.823	0.823
tfidf3-e	0.706	0.851	0.851	0.791	0.691	0.691
tfidf3-t	0.194	**1.000**	0.667	0.770	**0.936**	**0.936**
ngg-c3-e	0.926	0.862	0.862	0.722	0.561	0.561
ngg-c3-t	0.830	0.828	0.828	0.770	0.647	0.647
ngg-w3-e	0.360	0.711	0.569	0.766	0.740	0.740
ngg-w3-t	0.072	**1.000**	0.267	0.779	0.925	0.925
ngg-w1-et	0.909	0.863	0.863	0.799	0.691	0.691
mean	0.680	0.880	0.778	0.774	0.726	0.726

Table 2: Clustering results per configuration and metric for 20N and ML15. Larger values are better. Bold values indicate column maxima.

Comparing results on both datasets, it is noteworthy that the baseline TF-IDF models outperform their competition, with the second place occupied by both bag-based models (ML15) and graph-based ones (20N). Cluster merging yields good $PCPr$ scores (i.e., the penalized metric is identical to the non-penalized CPr). This is most prominent in the 20N dataset, where $PCPr$ equals CPr in all cases. In ML15, 30.7% of the examined methods fail to produce the correct number of clusters.

6. CONCLUSION

This study introduced a record linkage formulation to document clustering, adapting the definition and concepts to the new task. Using the JedAI toolkit, several experiments have been performed on two datasets, using various representation models, similarity measures, evaluation metrics and both raw document text and named-entities.

Results show that our approach performs well with respect to document clustering efficiency. The code used to run our experiments is available on GitHub[4]. Comparison with a brute force approach that does not adopt blocking techniques, shows the advantages of the record linkage pipeline, which map to speedups of *one and two orders of magnitude* with respect to the number of comparisons, with associated performance degradations of 5.6% and 24.4%, respectively.

The adaptation of the data-agnostic record linkage pipeline, combined with the availability of the modular and feature-rich JedAI toolkit, provides a plethora of interchangeable components to fine-tune document clustering performance in various domains, with minimal effort. However, a number of decisions on the entity linking level can heavily affect performance both in terms of effectiveness and (time) efficiency. Thus, it is in our next plans to propose case-specific blocking methods, to minimize the refinement needed in the pipeline while retaining effectiveness levels high. Such methods may take into account the implied entities, e.g. events vs. topics, and adjust the system tools applied accordingly.

7. REFERENCES

[Becker et al., 2011] Becker, H., Naaman, M., and Gravano, L. (2011). Beyond trending topics: Real-world event identification on twitter. *ICWSM*, 11(2011):438–441.

[Bi et al., 2016] Bi, X., Zhao, X., Ma, W., Zhang, Z., and Zhan, H. (2016). Record linkage for event identification in xml feeds stream using ELM. In *ELM-2015*, volume 1, pages 463–476. Springer.

[Brizan and Tansel, 2006] Brizan, D. G. and Tansel, A. U. (2006). A. survey of entity resolution and record linkage methodologies. *Communications of the IIMA*, 6(3):5.

[Daniel et al., 2003] Daniel, N., Radev, D., and Allison, T. (2003). Sub-event based multi-document summarization. In *HLT-NAACL 2003 Workshop on Text summarization*, volume 5, pages 9–16. ACL.

[Giannakopoulos, 2009] Giannakopoulos, G. (2009). *Automatic Summarization from Multiple Documents*. Ph. D. dissertation, University of the Aegean, Department of Information and Communication Systems Engineering.

[Giannakopoulos and Karkaletsis, 2009] Giannakopoulos, G. and Karkaletsis, V. (2009). N-gram graphs: Representing documents and document sets in summary system evaluation. In *TAC 2009*.

[Gomaa and Fahmy, 2013] Gomaa, W. H. and Fahmy, A. A. (2013). A survey of text similarity approaches. *International Journal of Computer Applications*, 68(13).

[Hassanzadeh et al., 2009] Hassanzadeh, O., Chiang, F., Lee, H. C., and Miller, R. J. (2009). Framework for evaluating clustering algorithms in duplicate detection. *VLDB 2009*, 2(1):1282–1293.

[Kuang et al., 2015] Kuang, D., Choo, J., and Park, H. (2015). Nonnegative matrix factorization for interactive topic modeling and document clustering. In *Partitional Clustering Algorithms*, pages 215–243. Springer.

[Kusner et al., 2015] Kusner, M., Sun, Y., Kolkin, N., and Weinberger, K. (2015). From word embeddings to document distances. In *ICML 2015*, pages 957–966.

[Papadakis et al., 2016] Papadakis, G., Svirsky, J., Gal, A., and Palpanas, T. (2016). Comparative analysis of approximate blocking techniques for entity resolution. *VLDB 2016*, 9(9):684–695.

[Papadakis et al., 2017] Papadakis, G., Tsekouras, L., Thanos, E., Giannakopoulos, G., Palpanas, T., and Koubarakis, M. (2017). JedAI: The force behind entity resolution. In *ESWC 2017*, pages 161–166. Springer.

[Reuter et al., 2011] Reuter, T., Cimiano, P., Drumond, L., Buza, K., and Schmidt-Thieme, L. (2011). Scalable event-based clustering of social media via record linkage techniques. In *ICWSM 2011*.

[Schenker et al., 2005] Schenker, A., Kandel, A., Bunke, H., and Last, M. (2005). *Graph-theoretic techniques for web content mining*, volume 62. World Scientific.

[Tsatsaronis et al., 2010] Tsatsaronis, G., Varlamis, I., and Vazirgiannis, M. (2010). Text relatedness based on a word thesaurus. *JAIR*, 37:1–39.

[Tsekouras et al., 2017] Tsekouras, L., Varlamis, I., and Giannakopoulos, G. (2017). A graph-based text similarity measure that employs named entity information. In *RANLP 2017*, pages 765–771.

[Wijaya and Bressan, 2009] Wijaya, D. T. and Bressan, S. (2009). Ricochet: A family of unconstrained algorithms for graph clustering. In *DASFAA 2009*, pages 153–167. Springer.

[4]https://github.com/npit/record-linkage