A Graph-based Text Similarity Measure That Employs Named Entity Information

Leonidas Tsekouras
Institute of Informatics and Telecommunications, N.C.S.R. "Demokritos", Greece,
ltsekouras@iit.demokritos.gr

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

George Giannakopoulos
Institute of Informatics and Telecommunications, N.C.S.R. "Demokritos", Greece,
ggianna@iit.demokritos.gr

Abstract

Text comparison is an interesting though hard task, with many applications in Natural Language Processing. This work introduces a new text-similarity measure, which employs named-entities' information extracted from the texts and the n-gram graphs' model for representing documents. Using OpenCalais as a named-entity recognition service and the JINSECT toolkit for constructing and managing n-gram graphs, the text similarity measure is embedded in a text clustering algorithm (k-Means). The evaluation of the produced clusters with various clustering validity metrics shows that the extraction of named entities at a first step can be profitable for the time-performance of similarity measures that are based on the n-gram graph representation without affecting the overall performance of the NLP task.

1 Introduction

The development of a text comparison algorithm is a critical step in many Natural Language Processing and Text Mining tasks, such as text clustering, categorization and summarization. However, the easy -for a human- task of understanding whether two texts are talking about the same topic or are somehow related, still remains an open challenge for NLP programs.

The main difficulties behind automatic text comparison are semantic ambiguity of words (Sanderson, 1994), lexical and syntactic differences (Ferreira et al., 2016) between sentences. According to Stavrianou et al. (2007), additional issues that affect text similarity performance and must be considered during text preprocessing are: stopwords and noisy data (e.g. misspelled words) removal, stemming, part of speech (POS) tagging, multi-word terms (collocations), tokenization and text representation. Text preprocessing in this direction aims at reducing the amount of information used for representing the document, only to the information that is really useful (e.g. by ignoring misspelled words or stopwords), by reducing semantic ambiguity (e.g. by defining the POS of a polysemous word) and the dimensions of the feature space (e.g. by mapping set of words to a multi-word term or by replacing words with stems).

Apart from the popular algebraic text representation model of VSM (Vector space model), where each word is a feature (Unigram or Bag-of-words model) and its multi-word (or multi-character) extensions (n-gram models), there has been significant work in representing texts as graphs. In the former cases, cosine similarity is used to calculate the similarity between two texts, whereas graph comparison methods are used in the latter case.

N-gram graphs (nGG) (Giannakopoulos and Karkaletsis, 2009; Giannakopoulos, 2009) capture the word order in the text, by connecting neighboring n-grams with edges that denote their frequency of co-occurrence within a given window of text and allow the detection of partial similarity in the morphology of text, with some resilience to noise and no need for preprocessing.Although n-gram graphs have shown improved performance in text mining tasks, their complexity significantly grows for large texts. In order to address this, we focus only on the most informative terms thus reducing the graph complexity, without losing significant information. It is typical in text representation models used in text mining or NLP tasks to select the most informative terms (e.g. the terms with the highest tf/idf weights or terms with special meaning, such as named entities). For example, Kumaran and Allan (2004) used Named Entity (NE)

Proceedings of Recent Advances in Natural Language Processing, pages 765–771,
Varna, Bulgaria, Sep 4–6 2017.
https://doi.org/10.26615/978-954-452-049-6_098

terms to improve performance in the "new event detection" task, Nadeau and Sekine (2007) provide an interesting survey on the uses of Named Entities in information extraction tasks, Toda and Kataoka (2005) employ Named Entities for clustering search results, whereas Sinoara et al. (2014) and Montalvo et al. (2015) used NEs as privileged information in text clustering.j

In this work, we combine the informativeness of Named Entities with the ability of the n-gram graph representation to capture word sequence information and define a new graph-based text similarity measure. We evaluate the performance of our approach in a text clustering task, using two different datasets. Results show that term selection can improve the time-performance of n-gram graph similarity and that named entities can be a useful addition to the set of terms selected using a "Term Frequency — Inverse Document Frequency" (TF-IDF) weighting scheme.

2 Related Work

In the past years, there has been significant research on text similarity. Gomaa and Fahmy (2013) provide a survey on text similarity measures, dividing them into three groups: i) string-based measures that operate on character sequences, ii) corpus-based measures that take into account information that comes from corpora, and iii) knowledge-based measures that use semantic networks (or similar knowledge-driven constructs) to determine the similarity between words.

Bos and Markert (2005) used surface string similarity, model building and theorem proving in order to assess text similarity. They extended the set of words in the text with synonyms from WordNet, and employed Google API to measure a weight for each word using the web as a corpus. Mihalcea et al. (2006) also used external information from semantic networks and defined a knowledge-based similarity metric for short texts, which reduced the error rate in a text paraphrasing task by up to 13% compared to other vector-based similarity metrics.

Friburger et al. (2002) found that the combined use of a "named entities" vector and an "all-words" vector with an increased weight to the entities vector had the best overall performance.

Schenker et al. (2005) performed text clustering and classifications tasks using graph representation models and graph-based similarity measures. They also introduced graph edit distance metrics, in order to tackle the complexity (NP-complete) of graph isomorphism problem. 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.

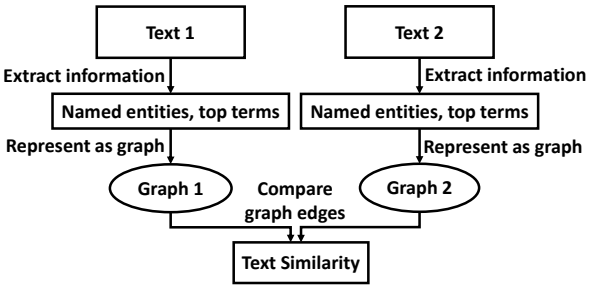

Figure 1: General diagram of how the algorithm works

The proposed approach builds on the metrics introduced by (Giannakopoulos and Karkaletsis, 2009) using the findings of (Friburger et al., 2002). Instead of taking all terms into account, we distinguish between named entities and top-ranked terms (by TF-IDF), and all other words and weight the n-gram graph accordingly.

3 Proposed Method

The aim of this work is to define a text comparison methodology that takes into account the named entities mentioned in the texts and represents texts as *n-gram graphs*, which are compared using graph comparison operators. For each pair of input texts T_i, T_j, a similarity function f will output a score $s = f(T_i, T_j)$, where $\{s \in \mathbb{R} | 0 \leq s \leq 1\}$ indicating how similar the two texts are. Values of s close to 1, indicate high similarity between the texts, when s is close to 0 the texts are dissimilar. The whole process is depicted in Figure 1.

In the information extraction step, two types of terms are extracted from text: i) named entities, ii) top-ranked terms using TF-IDF. The extraction of named entities has been done using the OpenCalais API[1], although any other entity extraction service or program can be used instead.

Using the named entities and the top terms extracted in the first step, we proceed to the text representation step, where: i) all entities are replaced with a hash value that allows multi-word entities to appear as single words in the word graph repre-

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

sentation model and ii) all the remaining words are replaced with a placeholder word. In the experiments, we chose the word "A" as a placeholder. The use of a single placeholder word causes the word graph to have only one node for all the non-important words, which significantly reduces the size of the n-gram graph and the complexity of comparison operators. Similarly, the mapping of the entity names to hash values minimizes the memory footprint of the graph further since a hash value takes up less memory than the full entity name in most cases.

Using the graph-based representation of the texts, the text similarity function is based on the comparison of the word n-gram graphs, which counts in tandem the value, size, containment and normalized value similarity of the two graphs as detailed in the following paragraphs.

3.1 Creation and Comparison of N-gram Graphs

For the creation of the word n-gram graph the JINSECT toolkit[2] has been employed, which supports both character and word n-gram graphs and implements several graph similarity measures. The word n-gram graphs are created using a sliding window of size n over the words, which means that a node is created for each word in the text (i.e. term hashcode or replacement word) and graph edges connect words (nodes) that are in proximity to each other (i.e. within a d words distance; we use $d = n$). The graph is weighted and weights denote the number of times two words were found close to each other (within the sliding window distance). For the comparison of two graphs, let's call them G_i and G_j, four similarity metrics that give a value in $[0, 1]$ have been employed. The metrics — Value Similarity, Size Similarity, Containment Similarity and Normalized Value Similarity — are defined by (Giannakopoulos, 2009) and for the comparison of a graph G_i against another graph G_j can be described as follows:

- *Value Similarity* indicates how many edges of G_i are present in G_j, but also takes into account the weights of these edges. If e is a given common edge of G_i, G_j with a respective weight of w_e^i, w_j^i, the we define $VR(e) = \frac{\min(w_e^i, w_j^i)}{\max(w_e^i, w_j^i)}$

$$VS(G_i, G_j) = \frac{\sum_{e \in G_i \cap G_j} VR(e)}{max(|G_i|, |G_j|)} \quad (1)$$

- *Size Similarity* takes into account only the size of the graphs.

$$SS(G_i, G_j) = \frac{min(|G_i|, |G_j|)}{max(|G_i|, |G_j|)} \quad (2)$$

- *Normalized Value Similarity* is assigned a value of 0, if Size Similarity is zero, otherwise it is the ratio of Value Similarity to Size Similarity. It is a measure of similarity that ignores the relative size of the graphs when comparing them.

$$NVS(G_i, G_j) = \frac{VS(G_i, G_j)}{SS(G_i, G_j)} \quad (3)$$

Below, we provide an example of a text that has been processed for extracting useful terms and has been represented as a word n-gram graph.

> **Original text:** ...Make your reservation early. Our **workshop** coincides with other **Cornell** events...
>
> **Processed text:** ...A A A A A WORKSHOP A A A -1675268131 A...

In addition, Figure 2 gives a visual representation of the word n-gram graph created by JINSECT for the example mentioned above, using a sliding window of size 3. The graph is quite small because all unimportant words are replaced with the same replacement word (i.e. "A") and consequently result to a single node in the graph. The edge going from node "A" to itself has a much bigger weight than other edges, because frequently in the text we have sequences of non-important words. The other nodes represent the named entities (the node with the hashcode value which corresponds to the entity Cornell) and top TF-IDF ranking words of the text (the top-1 word — Workshop — has been used). Both nodes are connected with the "A" node since they neighbor non-important words but not with each other.

The size of the word graph is small, because of the term extraction step. If the full text had been

[2] https://sourceforge.net/projects/jinsect/

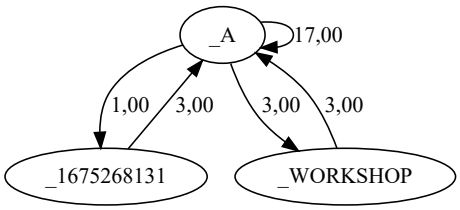

Figure 2: Example word graph with entities & top TF-IDF terms

used, then the graph would be larger and the time needed for graph and consequently text comparison would be larger. In the experiments, both the time complexity and the overall performance of the proposed methodology and of other methods are compared.

4 Experiments

This section describes the experimental evaluation process followed by an evaluation of the proposed text similarity measure performance in text clustering tasks. The datasets employed for the study are first presented (one English and one multilingual corpus), then the measures used for the evaluation of the cluster quality and finally the different steps of the processing pipeline, which have been evaluated for their time performance are explained. The presentation of results follows the same structure, starting with the clustering quality performance by dataset and continuing with the time complexity of the different tasks.

4.1 Datasets and Analysis Process

In the experiments, two datasets have been employed: i) the 20 Newsgroups data set, which consists of around 20,000 news documents, quite evenly distributed into 20 groups and ii) the MultiLing 2015 dataset, which comprises 1350 WikiNews articles in total about 15 events written in 10 different languages.

More specifically, for the 20 Newsgroups, we used the texts in the "test" set of the "bydate" version[3], which comprises documents from 20 different groups. Two texts have been excluded from the experiments, because they were in Swedish and this was not supported by the OpenCalais API and another five texts were excluded because they exceeded the maximum file size and maximum processing timeout set by the OpenCalais API. This resulted to a final set of 7525 texts.

In the case of the MultiLing 2015 dataset[4], which comprises texts derived from the publicly available WikiNews about various events, the dataset contains a number of events (15), each described by several documents (10–15). The documents have been translated across a number of languages. In our experiments, we used the English, Spanish and French versions, resulting in a set of 400 texts that cover the 15 events.

For evaluating the performance of the proposed text similarity measure in a text clustering task, we used a simple, centroid-based, clustering algorithm (i.e. k-Means) with a fixed number of clusters (k) that equals the number of predefined classes (i.e. $k = 20$ for 20 newsgroups, $k = 15$ for the MultiLing dataset). The input for the algorithm was a text similarity matrix, which was computed using i) the proposed similarity measure with entities only (ent_graph) and with top tf-idf terms ($ent\&tfidf\_graph$), ii) word n-gram graph similarity using the whole text to create the graph (all_graph), as defined by (Giannakopoulos and Karkaletsis, 2009), and iii) cosine similarity using the VSM representation and the top TF-IDF terms ($tfidf\_VSM$). Entities are extracted using OpenCalais and TF-IDF weights for words are computed using custom Java code. The n-gram graphs are constructed using the JINSECT library, using the replacement strategy described in section 3. The word n-gram graphs are compared using the JINSECT graph similarity metrics in order to create the document similarity matrix.

Using the text similarity matrix as input to k-Means, we produce a set of clusters. The ELKI software[5] has been employed for clustering the documents and more specifically the k-Means Lloyd implementation.

4.2 Clustering Validity Metrics

For the evaluation of clustering algorithms, the options are either to use external metrics that compare the clustering schema against a "ground truth" clustering or internal validity metrics that comparatively examine the cohesiveness and separation of clusters across different clustering schemata. In the current experiment, the "ground truth" is the actual classification of documents to the 20 newsgroups or the 15 MultiLing topics respectively, so external metrics are preferred. Since

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

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

[5] https://elki-project.github.io/

all experiments have been done using the same clustering algorithm (k-Means), the only factor that affects the cluster quality is the text similarity measure, so the results are directly comparable.

For measuring the validity of the produced clustering schemes, we implemented a wide range of external clustering quality indexes: Precision, Recall and F_1-measure as described by (Hassanzadeh et al., 2009), Folkes and Mallows ($F\&M$), Jaccard and Rand as described in (Desgraupes, 2013).

Since k-Means' results depend on the selection of the initial k centroid documents, we repeat the clustering many times (100 for the MultiLing dataset and 10 for the much larger 20-newsgroups dataset) and we report the mean values and the 95% confidence intervals in Tables 1 and 2.

4.3 Time Performance

The whole pipeline of information extraction (entities and TF-IDF weights), text representation (as graphs or vectors), text similarity computation and clustering was wrapped in a Java program that employs the JINSECT library, the OpenCalais API and the ELKI clustering algorithms. This allows to measure the time needed for the different steps of the comparison procedure:

TF-IDF weights refers to the time needed for the computation of TF-IDF weights for all the texts in the dataset. In the case of word n-gram graphs created using the whole text, this step is omitted.

Graph creation is the time needed for the creation of all word n-gram graphs (one for each text). The graphs are cached in memory in order to accelerate the steps that follow.

Graph comparison is the time needed to create the similarity matrix, containing the pairwise similarities of all texts in the dataset.

We do not report the time for extracting named entities, since it involves accessing the external OpenCalais API, and time performance depends on factors that cannot be controlled, such as the network latency. In the future, we aim to replace this step with an offline Named Entity Recognition service based on the open source OpenNLP project. We do not also report the time for running the clustering algorithm or for evaluating the clustering results, since it is expected to be equivalent in all cases, given the fixed size of the similarity matrices.

4.4 Results

The quality of the clusters produced by k-Means, using the four different similarity measures (i.e. the baseline cosine similarity that uses TF-IDF — $tfidf\_VSM$, the n-gram graph similarity using all words — all_graph, the proposed n-gram graph similarity measure with entities only — ent_graph, and an extension that combines entities and the top TF-IDF terms — $ent\&tfidf\_graph$) has been evaluated using the validity indexes.

4.4.1 English Texts (20 Newsgroups)

The results for 20 Newsgroups are summarized in Table 1. Results in bold are significantly better (at 95% confidence interval) than that of the baseline $tfidf - VSM$ method, which employs cosine similarity and the TF-IDF weighting scheme. All the approaches demonstrate a rather low performance (at least in Precision and F_1 Measure), which is mainly due to the large number of categories and the sparsity of the unigram (i.e. words) feature space (less than .5% non-zero features). This sparsity is even greater in the case of entities, where only 3% of the document pairs have at least one common entity. This explains the poor performance of the ent_graph method, which still outperforms the original all_graph method and the $tfidf\_VSM$ baseline.

The approach that adds to the entities n-gram graph a few more nodes that correspond to important document words (high TF-IDF values) improves the results significantly (all values have been computed at the 95% confidence interval) against the VSM and the simple n-gram graph model, but also outperforms the graph based similarity that uses only the entities in the graph. In this case, the important document terms increase the overlap between the document graphs.

	$tfidf\_VSM$	all_graph	ent_graph	$ent\&tfidf\_gr.$
$F\&M$	**0.15 ± 0.012**	0.10 ± 0.012	0.14 ± 0.019	0.09 ± 0.005
Jaccard	0.05 ± 0.001	0.04 ± 0.002	0.05 ± 0.001	0.04 ± 0.002
Rand	0.58 ± 0.067	0.81 ± 0.044	0.65 ± 0.099	**0.87 ± 0.016**
Precision	0.06 ± 0.005	0.06 ± 0.009	0.08 ± 0.017	**0.10 ± 0.017**
Recall	**0.61 ± 0.075**	0.32 ± 0.077	0.52 ± 0.122	0.24 ± 0.037
F_1	0.10 ± 0.008	0.10 ± 0.008	0.13 ± 0.020	**0.14 ± 0.012**

Table 1: Clustering performance for the 20 Newsgroups dataset (95% C.I.)

4.4.2 Multilingual Texts (MultiLing)

The results are even more interesting in the case of the multi-lingual dataset of MultiLing and are

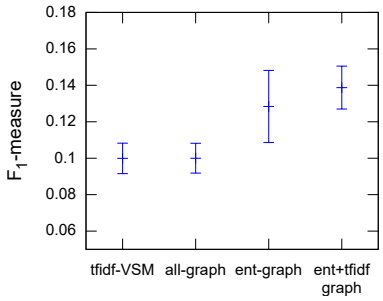

Figure 3: F_1-measure performance of the algorithms for the 20 Newsgroups dataset

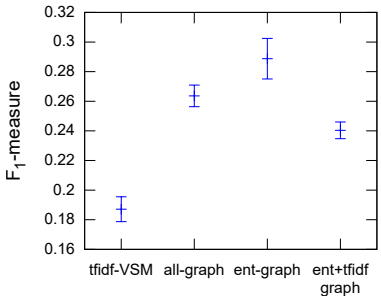

Figure 4: F_1-measure performance of the algorithms for the MultiLing 2015 dataset

summarized in Table 2. In this dataset, the ent_graph method significantly outperforms all other methods (in terms of Recall and F_1), and the entities seem to be more useful than all the other words in the texts. These results highlight one of the main advantages of the proposed graph-based similarity measure, which is the ability to process texts in multiple languages. The entity extraction mechanism reduces the feature space only to the named entities, which frequently remain the same between languages, thus reduce sparsity. In this dataset, 38% of the document pairs had at least one common entity.

What is also interesting here is that the performance degrades when important document terms (according to TF-IDF) are added to the graph. This is because such terms are translated across languages are not matched thus reduce the similarity of the corresponding document graphs.

	$tfidf\_VSM$	all_graph	ent_graph	$ent\&tfidf\_gr.$
$F\&M$	0.16 ± 0.001	0.16 ± 0.001	$\mathbf{0.21 \pm 0.003}$	0.15 ± 0.002
Jaccard	0.08 ± 0.001	0.08 ± 0.001	0.08 ± 0.002	0.08 ± 0.001
Rand	0.77 ± 0.006	0.82 ± 0.006	0.62 ± 0.021	$\mathbf{0.85 \pm 0.003}$
Precision	0.13 ± 0.008	0.23 ± 0.010	0.20 ± 0.013	0.21 ± 0.008
Recall	0.35 ± 0.008	0.33 ± 0.005	$\mathbf{0.62 \pm 0.020}$	0.29 ± 0.004
F_1	0.19 ± 0.008	0.26 ± 0.007	$\mathbf{0.29 \pm 0.014}$	0.24 ± 0.006

Table 2: Clustering performance for the MultiLing 2015 dataset (95% C.I.)

4.4.3 Time Complexity

Figure 3 presents the time for the various text comparison steps for the 20 Newsgroups dataset only, since the respective times for the 400 texts of the MultiLing 2015 dataset where very small. The time for graph construction for the 7525 texts almost doubles when all words are used whereas time for text comparison almost triples.

The TF-IDF time for $tfidf\_VSM$ is longer than that of $ent\&tfidf\_graph$'s because we include the time required to create vectors of the same size for each document, which is required to calculate their cosine similarity but not to just identify a document's top terms.

	all_graph	$ent\&tfidf\_graph$	$tfidf\_VSM$
TF-IDF	0	3.97	26
Graph creation	38.46	17.6	0
Text comparisons	1697.3	509.4	543.2

Table 3: Times for the various text comparison steps (in seconds)

5 Conclusion

This work presented a graph-based text similarity measure that takes advantage of named entities' information and improves the performance of text clustering tasks. The similarity measure employs named entities and the most important document terms (by TF-IDF) for the construction of the n-gram graph and improves the time complexity of the n-gram graph similarity measures that employ all the document information since it results in a smaller and simpler graph. The first results show that the method can be useful in cases where the documents are rich in entities and have an overlap in the entities space, and is not very useful in the absence of entities. The proposed measure is appropriate for multilingual text collections (e.g. for news collected in many different languages), since the named entities seem to be less affected by translation than any other word in the text.

It is on our plans to evaluate the performance of our entity based similarity measure to character n-gram graphs, which are expected to capture better the small variations across languages. Even, when named entities are translated from a language to another, the differences are small and could be possibly captured by a character n-gram graph model.

References

Bos, J. and Markert, K. (2005). Recognising Textual Entailment with Logical Inference. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 628–635, Stroudsburg, PA, USA. Association for Computational Linguistics.

Desgraupes, B. (2013). Clustering indices. *University of Paris Ouest-Lab ModalX*, 1:34.

Ferreira, R., Lins, R. D., Simske, S. J., Freitas, F., and Riss, M. (2016). Assessing sentence similarity through lexical, syntactic and semantic analysis. *Computer Speech & Language*, 39:1–28.

Friburger, N., Maurel, D., and Giacometti, A. (2002). Textual similarity based on proper names. In *Proc. of the workshop Mathematical/Formal Methods in Information Retrieval*, pages 155–167.

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

Giannakopoulos, G. and Karkaletsis, V. (2009). N-gram graphs: Representing documents and document sets in summary system evaluation. In *Proceedings of Text Analysis Conference TAC2009 (To appear)*.

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

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

Kumaran, G. and Allan, J. (2004). Text classification and named entities for new event detection. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 297–304. ACM.

Mihalcea, R., Corley, C., and Strapparava, C. (2006). Corpus-based and knowledge-based measures of text semantic similarity. In *AAAI*, volume 6, pages 775–780.

Montalvo, S., Martínez, R., Fresno, V., and Delgado, A. (2015). Exploiting named entities for bilingual news clustering. *Journal of the Association for Information Science and Technology*, 66(2):363–376.

Nadeau, D. and Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

Sanderson, M. (1994). Word sense disambiguation and information retrieval. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 142–151. Springer-Verlag New York, Inc.

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

Sinoara, R. A., Sundermann, C. V., Marcacini, R. M., Domingues, M. A., and Rezende, S. O. (2014). Named entities as privileged information for hierarchical text clustering. In *Proceedings of the 18th International Database Engineering & Applications Symposium*, pages 57–66. ACM.

Stavrianou, A., Andritsos, P., and Nicoloyannis, N. (2007). Overview and semantic issues of text mining. *ACM Sigmod Record*, 36(3):23–34.

Toda, H. and Kataoka, R. (2005). A search result clustering method using informatively named entities. In *Proceedings of the 7th annual ACM international workshop on Web information and data management*, pages 81–86. ACM.