Grouping business news stories based on salience of named entities

Llorenç Escoter, Lidia Pivovarova, Mian Du, Anisia Katiskaya and Roman Yangarber

Department of Computer Science
University of Helsinki, Finland

first.last@cs.helsinki.fi

Abstract

In news aggregation systems focused on broad news domains, certain stories may appear in multiple articles. Depending on the relative importance of the story, the number of versions can reach dozens or hundreds within a day. The text in these versions may be nearly identical or quite different. Linking multiple versions of a story into a single group brings several important benefits to the end-user—reducing the cognitive load on the reader, as well as signaling the relative importance of the story. We present a grouping algorithm, and explore several vector-based representations of input documents: from a baseline using keywords, to a method using *salience*—a measure of importance of named entities in the text. We demonstrate that features beyond keywords yield substantial improvements, verified on a manually-annotated corpus of business news stories.

1 Introduction

We address the problem of grouping multiple versions of the same story in a system that continuously processes articles incoming from a large number of news streams. Our problem setting is PULS—an on-line information extraction (IE) system, which analyses news in the business domain.[1] PULS collects articles from over 1000 on-line sources that provide RSS feeds. Among approximately 4000–6000 articles arriving daily, some of the stories appear multiple times.

The role of the aggregation component is to cluster the articles into a set of *stories*—a story is a set of articles that report the same piece of news. The purpose of grouping is that when a user issues a query, the system should show one item per story, rather than one item per article, so the same fact is not shown over and over again. Information should be presented in a way that minimizes redundancy of the returned results; this implies narrowly clustering news articles that describe the same events or facts.

Another goal is to identify *trending* stories. In the business domain, if a story is globally important, it will appear in many feeds. When repeated news items are identified, the number and variety of sources covering the story is an indicator of that story's relative importance.

Thus, from the end-user's perspective, we have at least two motivations for grouping different news that describe the same story: reducing redundancy and ranking stories by importance.

The main contributions of this paper are:

- We try to identify the most effective document representation for news clustering. We demonstrate that automatically extracted *named entities* (NE) are better features than words. Moreover, considering the *salience* of NEs—a measure that combines frequency and prominence of the NE—gives further improvement in clustering performance. We introduce a novel salience weighting scheme, which in our experiments outperforms TF-IDF and raw count weighting.
- For word representation, we compare pre-trained *word2vec* vectors with vectors trained on a domain-specific news corpus. Although the corpus-specific word embeddings alone give lower performance on the clustering task, we show that they work better in combination with NE features.
- We analyze two measures for evaluation of clustering performance—Rand index and V-

[1] http://puls.cs.helsinki.fi/

measure—and a standard way of adjusting them for "chance." We demonstrate that adjusting favors clustering with a smaller number of clusters. We propose a new type of normalization, which avoids this problem.

- We publish the *data-set*, consisting of nearly 4000 articles collected by our system for one day, grouped into clusters manually. The data-set represents a real task, which the aggregation component must solve daily. We also provide a command-line *annotation tool*, to facilitate manual clustering. We also publish the *word embeddings*.

Although in this paper we deal only with business news, we consider redundancy and the tendency to repeat the more important events to be general properties of news streams. Thus, we believe that our task and the proposed solution generalize to many other news-monitoring settings.

The paper is organized as follows. Section 2 discusses related work. Section 3 presents the NE extraction system and introduces salience for NE weighting. Section 4 describes the algorithm and features. Section 5 describes the data and annotation process. Section 6 discusses evaluation methods and results. Section 7 contains conclusions and future work.

2 Related work

A general overview of text clustering techniques can be found in (Aggarwal and Zhai, 2012). Many results on document clustering are published in IR literature, where these techniques are used to cluster search results (Carpineto et al., 2009).

News are clustered for various purposes: finding breaking news in streams (Kumaran and Allan, 2004), linking duplicates or articles about the same story (Vadrevu et al., 2011), tracking threads of news over time (van Erp et al., 2014; Azzopardi and Staff, 2012; Steinberger and Pouliquen, 2008), or facilitating access to information (Zhang et al., 2013; Toda and Kataoka, 2005).

The main techniques for clustering documents are: agglomerative clustering (Steinberger and Pouliquen, 2008) and partitioning clustering, such as k-means, buckshot, and *fractionation* (Azzopardi and Staff, 2012; Sankaranarayanan et al., 2009; Cutting et al., 1992). Hierarchical agglomerative clustering is commonly used in practice, though in general it has a complexity of $O(n^2 \log(n))$, (Berkhin, 2006). All objects start in their own, trivial cluster. The closest pair of clusters is merged, iteratively, until the hierarchy is complete. Partitional algorithms can also be used to create a hierarchical solution, e.g., *bisecting k-means*, which is better than standard k-means and comparable to agglomerative hierarchical approaches, (Steinbach et al., 2000).

On the other hand, determining the number of clusters might be a tricky task for partitional algorithms (Gialampoukidis et al., 2016).

Suffix Tree Clustering (STC) is a linear-time algorithm based on identifying common phrases within groups of documents, (Zamir and Etzioni, 1999). Spectral clustering models the documents as an undirected graph, where each node represents a document, assigns a similarity between documents as a weight on the edges, and tries to find the best cuts of the graph, (Shi and Malik, 2000). Xu et al. (2003) identify document clusters in the latent semantic space derived by non-negative factorization of the term-document matrix of the given corpus.

A common procedure for agglomerative clustering (also used in this paper) can be summarized as follows: convert documents into a vector representation, then use a metric to compute pairwise similarity between documents—often, cosine similarity. In this procedure, clustering quality crucially depends on document representation.

Traditionally, a common way of representing documents for clustering is by a vector of TF-IDF weights for each keyword, e.g., (Iglesias et al., 2016; Azzopardi and Staff, 2012; Vadrevu et al., 2011). Steinberger and Pouliquen (2008) use log-likelihood (LL) for weighting keywords rather than TF-IDF. LL statistics can be computed for each word in the corpus, relative to a separate, "reference" corpus (Rayson and Garside, 2000). Staff et al. (2015) claim that for search results, using raw term frequencies outperforms TF-IDF. Recent lines of research use word embeddings (Mikolov et al., 2013b) to represent documents. Sophisticated deep learning algorithms can also be applied to text clustering (Xu et al., 2015), but to date they require labeled training data, while the method proposed in this paper is unsupervised.

In contrast to bag-of-words (BOW) schemes, named entities (NEs) can be used as features (Montalvo et al., 2012). In most cases, NEs are also weighted according to TF-IDF (Toda and Kataoka, 2005) or its variants (Cheng et al., 2012;

Kiritoshi and Qiang, 2016).

Kumaran and Allan (2004) combined three vector representations for a document, namely: all words, NEs, and all words except NEs. This is similar to the series of experiments in this paper; the difference is that Kumaran and Allan (2004) used TF-IDF in all three cases, while we compare TF-IDF with several alternatives.

Popular clustering data sets target much coarser categorization tasks. For instance, 20 Newsgroup[2] and Reuters' RCV1[3] categorize news into business sectors such as "Fruit Growing" or "comp.graphics." TDT2[4] classifies news related to certain topics—a phenomenon or a big event– such as "Asian Economic Crisis" or "1998 Winter Olympics." Our task focuses on more focused groups; stories about business activities that occurred within a given industry sector or that are related to a broader phenomenon are not considered the same story. If the same entities engage in two different activities, we consider that as two distinct stories. Therefore, we manually annotated a sample of our corpus, which is more suitable for evaluating our methods.

3 Named Entities and Salience

We use a Named Entity Recognition module as part of the PULS news monitoring system. (Yangarber and Steinberger, 2009; Huttunen et al., 2012; Du et al., 2016) It uses patterns and rules to extract NEs; currently the system uses about a thousand patterns, some of which were learned (Yangarber, 2003) and some manually constructed. The system assigns a *type* to each NE—company, person, product, etc.—but the NE types are not used for grouping to reduce the effect of mistakes in analysis, e.g., when an entity is classified with different types across multiple documents. Rather, we consider clustering to be an earlier step in the overall processing pipeline (Yangarber, 2006). Ji and Grishman (2008) show that performance of an IE system can be improved by using clusters of topically-related documents. In PULS we use grouping to improve NE classification: we assign each entity a type based on the majority within the set of clustered documents.

[2] http://www-2.cs.cmu.edu/~TextLearning/datasets.html

[3] http://about.reuters.com/researchandstandards/corpus/

[4] https://catalog.ldc.upenn.edu/LDC2001T57

Our definition of *salience* relies on the general nature of news articles. Authors typically mention the main event in the title, in condensed form; then, the main information is elaborated in the first few sentences, followed by further detail and background. Thus, the most important NEs are mentioned early in the text and then repeated, whereas less important NEs are mentioned in the later paragraphs and are less frequent.

We compute *salience* as a combination of *prominence* and *frequency* of an entity in a document. Prominence captures the importance of the *first* mention of entity e in document d:

$$\textit{prominence}(e,d) = \frac{\text{ns}(d) - \text{fs}(e,d)}{\text{ns}(d)}$$

where $\text{ns}(d)$ is the total number of sentences in the document, $\text{fs}(e,d)$ is the number of the sentence (starting at zero) of the first mention of e in d. Thus, the prominence of entities mentioned in the title is 1. Prominence also takes into account the total length of the document, to capture diversity of news sources in the collection. For example, the second sentence in a two-page article is more important than the second sentence in a short article, where all crucial information must be condensed at the very beginning.

Frequency is the ratio of mentions of a given NE over all NE mentions:

$$\textit{frequency}(e,d) = \frac{C(e,d)}{\sum\limits_{e' \in \text{NE}(d)} C(e',d)}$$

where $C(e,d)$ is the count of e in d, $\text{NE}(d)$ is the set of all NEs in d. Note that we compute the NE frequency relative to the other NEs only and ignore all the other words in the document, since NEs and common words have rather different distributions: important terms are usually repeated more times than names.

We define salience as the geometric mean of prominence and frequency:

$$S(e,d) = \sqrt{\textit{prominence}(e,d) \cdot \textit{frequency}(e,d)}$$

Salience lies between 0 and 1, but the saliences in a document need not add up to one—there may be more than one salient entity in the document, or none. In the business domain, the majority of events involve some NEs (often, companies).

We make extensive use of salience in the PULS system to aggregate and present information to

users. For example, we represent each group as a list of salient companies, with the least salient companies removed. Similarly, when a user searches for a name, the system returns only documents where the entity salience is above a certain threshold.

We compare salience to two other weighting strategies: namely, frequency alone, and TF-IDF.

4 Clustering algorithm

The experiments follow the same structure. We start with a collection of documents and transform it into a collection of vectors, by one of the methods described below. We apply agglomerative clustering to the collection of document vectors, using cosine distance between vectors. The agglomerative algorithm produces a dendrogram with the documents as leaves, and we obtain a clustering by cutting at a distance threshold $\theta \in (0, 1)$. We use the *complete* metric, meaning that the distance between clusters A and B is the *maximum* of the distances between any two vectors in A and B (Aggarwal and Zhai, 2012). The threshold θ imposes a limit on the maximum distance between any two documents in a cluster.

4.1 Mapping documents to $\mathbb{R}^k$

To represent documents as vectors we use two types of features: all words, or named entities. For words, we use three representations: TF-IDF, *word2vec* embeddings pre-trained on a large general corpus, and embeddings trained on our business-news corpus. For NEs, we use raw counts, TF-IDF and salience. Thus, we experiment with six vector representations.

Word-based representations: For each word w, we compute TF-IDF as:

$$\text{TF-IDF}(w, d) = \frac{C(w, d)}{\sum_{w' \in W(d)} C(w', d)} \cdot \log \frac{|D|}{|D_w|}$$

where $C(w, d)$ counts how many times the word w appears in document d, $W(d)$ is all words in d, D is all documents in the corpus, and $|D_w|$ are all documents in D which contain word w. Then the vector representation for the document is:

$$\text{TF-IDF}(d) = \sum_{w \in W(d)} \text{TF-IDF}(w, d)\hat{u}_w$$

where $\hat{u}_w$ is the *one-hot* vector, whose length is the size of the vocabulary, and which contains zeros in all positions but the one corresponding to w. The vector TF-IDF(d) contains TF-IDF values for its words and zeros in all other positions. We use only content words—nouns, adjectives and verbs—in the TF-IDF representations.

w_1	w_2	Google News	Business
jump	climb	0.55	**0.85**
recall	remember	**0.43**	0.13

Table 1: Cosine similarity for sample word vectors.

Another approach is to represent each word as a vector in a low-dimensional vector space. We can then represent documents by adding their corresponding word vectors. We use word vectors produced by the CBOW approach—continuous bag-of-words (Mikolov et al., 2013a). The vector representing document d is then:

$$\text{CBOW}(d) = \sum_{w \in W(d)} C(w, d)e_w$$

where e_w is the *embedding* vector representing w.

In this paper we use the "standard" word2vec embeddings built on the Google News data-set[5] (referred to as "CBOW-st"), and embeddings trained on our business-news corpus "CBOW-b". Our corpus is relatively small (4.5 million documents), but it contains only documents relevant to business news. We do not know *a priori* which set of embeddings is more suitable for our task. Although the two embeddings produce similar results, the resulting word vectors have noticeable differences, as can be seen in Table 1. The business-domain embeddings for *jump* and *climb* are much closer than in the general corpus, since both are used to denote *increases*; meanwhile, embeddings for *recall* and *remember* are much closer in the general corpus, because, in the business domain, *recall* frequently refers to product recalls.

Named entity-based representations: Another natural representation for a document d can be obtained by using only named entity counts:

$$\text{NEC}(d) = \sum_{e \in \text{NE}(d)} C(e, d)\hat{u}_e$$

where NE(d) is the set of all named entities in document d,[6] and $\hat{u}_e$ is the one-hot vector. TF-IDF for NEs is computed in the same way as for keywords

[5] code.google.com/archive/p/word2vec

[6] Here we use counts instead of frequencies since there are equivalent when cosine similarity is used.

but using only NEs, ignoring common words. We refer to this representation as **NE-TFIDF**. Finally, a salience-based document representation leverages the *salience* S of the NEs, described in Section 3:

$$\text{NES}(d) = \sum_{e \in \text{NE}(d)} S(e,d)\hat{u}_e$$

4.2 Combining representations

Named entities are crucial for news clustering. In some reporting, journalists use standardized language and even article templates to describe similar events. In such cases, the article's NEs are the only way to obtain a meaningful document similarity. On the other hand, NEs can be misleading, e.g., if two different events take place in the same location (see (Kumaran and Allan, 2004) for examples in both kinds of problems). Thus, we try to *combine* NE and other textual features.

To that end, we use the best-performing methods in their respective categories—see results in Section 6—namely CBOW for words and salience for NEs. We use two methods of combining of CBOW and NES representation: *juxtaposition* and the *AND* function.

Juxtaposing means simply appending together the vectors corresponding to the NEs and the keywords, to form longer document vectors:

$$\text{NES\_CBOW}_\alpha(d) = \left[\alpha k_d \frac{\text{NES}(d)}{\|\text{NES}(d)\|}, \frac{\text{CBOW}(d)}{\|\text{CBOW}(d)\|}\right] \quad (1)$$

where first, we normalize both representations by their respective Euclidean distances, $\|\cdot\|$; then, we scale by α, which controls the relative weight of NES vs. CBOW. We further scale NES by k_d—the number of NEs found in d. The rationale behind this is that if a document contains more NEs, then the NES representation conveys more information; whereas if d has only one NE, then grouping should rely much more on the keywords. We apply the same agglomerative clustering procedures as in other experiments to these juxtaposed vectors. We experiment with both vector representations, CBOW-st and CBOW-b.

The second method, similar to that used in (Kumaran and Allan, 2004), requires that both word distance *and* NE distance should be sufficiently close—closer than corresponding thresholds. In this case, we cannot use the *complete* linkage metric since a maximum of distances is not defined if the distance is a *pair* of numbers. Thus, unlike all other experiments described in this paper, for the *AND* combination method we use the *single* metrics (Aggarwal and Zhai, 2012). This is equivalent to finding connected components of the graph where the nodes are documents and there is and edge between two nodes if and only if both distances are below their respective thresholds.

5 Data and annotation scheme

From our business news corpus (Pivovarova et al., 2013) we selected one "typical" day for annotation, with a total of 3959 documents.[7] We manually annotated all of these documents via a specialized interface, which displays documents pairwise and allows an annotator to make three main decisions: documents can be

- Grouped: if their main stories are the same.
- Not grouped: if their stories are not the same.
- Partially grouped: if their main stories are not the same, but may partially overlap. For example, one article might mention the other's main story toward the end.

The interface provides other helpful options, for example, the annotator can use regular expressions to search for all documents similar to a given one; another option is to mark the document as invalid if the document is *malformed*.[8]

We use a triangular matrix M to keep track of the pairwise relations among documents. Since annotation is an extremely time-consuming process, the key aim is to reduce the amount of data shown to the annotator as far as possible. We initialize M by pre-marking all pairs of documents that do have no NE in common as un-grouped; this decision may later be reversed by the annotator.

Another idea that allows us to minimize manual work is *decision propagation*. Document grouping—that is, documents having the same main story—can be viewed as an equivalence relation. This means that (ideally) only one member of a group needs to be checked against a member of another group to decide whether both groups

[7] Some sites publish "summary" articles, which contain an overview of 10 or more (possibly unrelated) stories, with as little as one sentence per story. In this paper, summaries are filtered out, to make the grouping task well defined. Filtering is performed by a simple segmentation algorithm, which checks whether the text is separable into contiguous segments, containing *non-overlapping* sets of named entities.

[8] This includes documents where some *broken* content was retrieved, such as a login page or advertisement.

should be merged. Every time a user groups two documents, the decision is propagated so that two corresponding groups are merged, and other pairs from the merged group are never shown to the user. Sometimes this leads to a contradiction, including cases when the initialization was wrong—i.e., when initialization suggests that two documents cannot be in the same group, but the annotator decides to merge two groups they belong to. In such a case, the annotator is warned and has to resolve the contradiction. Negative decisions can also be propagated: when an annotator marks a pair of documents as ungrouped, this affects all documents in both groups. Partial relations, on the other hand, are not equivalence relations, and require more manual annotation.

Our annotation tool keeps track of annotators' decisions, U, and reconstructs the annotation from them. This process can be viewed as applying each $u \in U$ to successive versions of M:

$$M_i = u_i(M_{i-1})$$

In this scenario, mistakes—which will appear as contradictions in M—are much easier to detect and correct. Given U that generates contradictions, a minimal subset $U' \subset U$ that generates the same contradiction can be more easily inspected. Then, the offending input can be corrected and we can proceed with the annotation.[9]

In total four members of our team were involved in the annotation process. Most of the instances were annotated by one person. In the beginning of the process we annotated several cases together and discussed difficult ones to work out general guidelines. Annotators also checked some random part of others' annotations, and corrected several cases during error analysis, by looking at misclassified instances with the highest confidence.[10]

Of the 3959 documents annotated in this manner, all documents marked as either invalid or *partially related* to others were removed (402 documents), leaving 3557 documents that can be grouped unambiguously. This constitutes the ground-truth clustering against which we test our system. Figure 1 shows how the documents are distributed among cluster sizes: the vast majority of them (2249) are in a cluster by themselves,[11]

[9]We arrived at this annotation scheme through trial and error, since annotating thousands of documents is a complex and tedious task. This seems to be an effective approach.

[10]Overall, the annotation process spanned across two calendar months.

[11]Leftmost bar is cut off at 500 to improve readability.

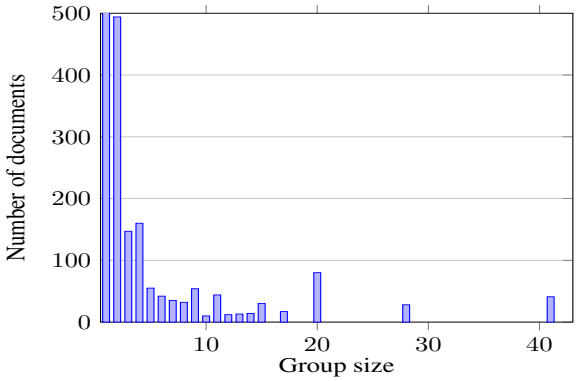

Figure 1: Distribution of annotated data (the leftmost bar goes up to 2249: 2249 documents are not clustered with any other document)

and the rest form larger ones. This is expected, as most of the data has been gathered from specialized business RSS feeds (mining news, dairy news, and so on); these sources usually report all news related to their industry including less important events that do not appear in other sources.

6 Evaluation and results

6.1 Evaluation methods

We evaluate our resulting clusters using Rand index (RI) and V-Measure. Rand index considers all possible pairs of documents, and is the proportion of correctly classified pairs—grouped or ungrouped—among all pairs (Rand, 1971). RI can be adjusted for chance, as described in (Hubert and Arabie, 1985):

$$ARI(S_{\text{rep}}) = \frac{RI(S_{\text{rep}}) - \mathbb{E}[RI(S_{\text{chance}})]}{1 - \mathbb{E}[RI(S_{\text{chance}})]} \quad (2)$$

where S_{rep} is the clustering strategy based on a document representation rep—rep is one of the representation strategies described in Section 4. The strategy S_{chance} is a *random* clustering strategy; $RI(S)$ is the RI of applying strategy S to the data; and $\mathbb{E}$ is expectation, which is estimated as described in (Hubert and Arabie, 1985).

V-Measure is the harmonic mean of homogeneity (H) and completeness (C) (Rosenberg and Hirschberg, 2007).

$$H = 1 - \frac{\mathbb{H}(C|K)}{\mathbb{H}(C)} \quad C = 1 - \frac{\mathbb{H}(K|C)}{\mathbb{H}(K)} \quad V = \frac{2HC}{H + C}$$

where $\mathbb{H}$ denotes entropy, K are predicted labels,

and C are the true labels. Completeness, homogeneity and V-measure are analogous to recall, precision and F-measure respectively.

Figures 2 and 3 show the measures we wish to optimize, namely the Rand index (*unadjusted for chance*) and the V-Measure. RI adjusted for chance (ARI) is shown in Figure 4.

From figures 2 and 3 it is relatively difficult to see the maxima of the measures. In both, most of the *interesting* information lies in a very small region, which is difficult to visualize. If we zoomed into these regions sufficiently, we would find that the maxima in these figures correspond to similar values θ, which are different from the maxima in Figure 4. This highlights several problems.

Adjustment for chance depends on the number of clusters, which in turn depends on the threshold value (θ); a random clustering that produces a very large number of clusters will have a very high RI value (which is the adjustment penalty) and the ARI will be very low, because the penalty for the adjustment is high. Therefore, the ARI measure favors higher values of θ, where the number of resulting clusters will be smaller.

We consider this a shortcoming of the ARI measure and propose another function to maximize. We rather adjust for the **naïve** strategy, $S_{\text{naïve}}$, which assigns each document to its own cluster. In other words, we try to measure what is the gain of clustering some documents compared to "doing nothing."

We transform Equation 2 to adjust for $S_{\text{naïve}}$ rather than chance. Suppose $f(S)$ is a scoring measure for a clustering strategy S; in our case, f is RI or V-measure. Now, $f(S) \in [0, 1]$, and 1 is the perfect score, and 0 is the worst score. We can adjust f as follows:

$$\begin{aligned}\hat{f}_{\text{naïve}}(S_{\text{rep}}) &= \frac{f(S_{\text{rep}}) - \mathbb{E}[f(S_{\text{naïve}})]}{1 - \mathbb{E}[f(S_{\text{naïve}})]} \\ &= \frac{f(S_{\text{rep}}) - f(S_{\text{naïve}})}{1 - f(S_{\text{naïve}})} \end{aligned} \quad (3)$$

Equation 3 adjusts the score for the naïve strategy—which shows how much better than the naïve the given strategy performs; if it performs worse than naïve adjusted score is less than zero.

The naïve strategy produces high scores: V-measure of 0.965 and RI of 0.9993. Homogeneity for the naïve strategy is 1, and completeness for our corpus is also quite high because the majority of documents do not belong to any group, as shown in Figure 1. Rand index considers all possible pairings and yields a high score since most must belong to different clusters.

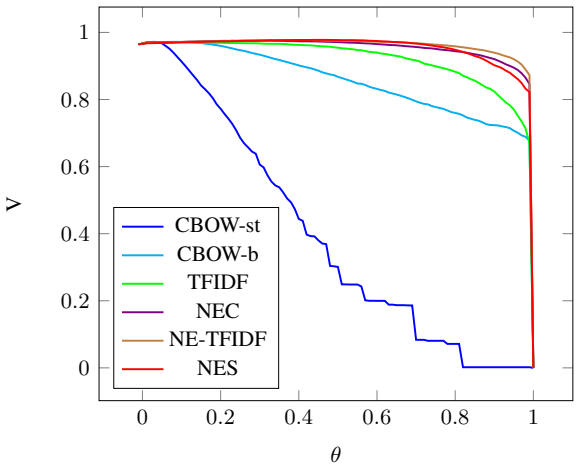

Figure 2: V-Measure

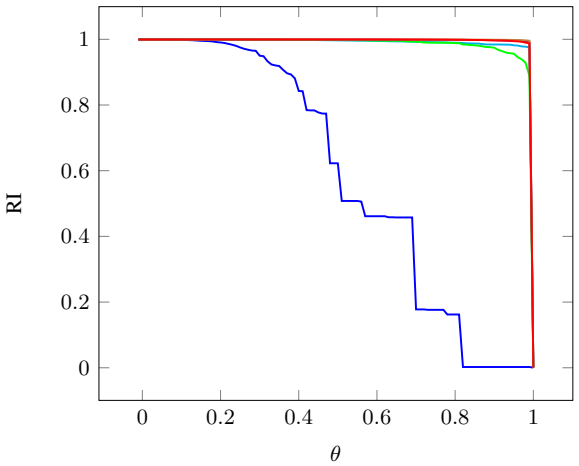

Figure 3: Rand index

Using the naïve adjustment, Figures 5 and 6 show a much clearer picture of how each document representation behaves.[12] The figures show that the two measures—RI and V-measure—behave similarly, and reach their maxima at very similar values of θ. Because the measures indicate the same maximum, we do not need to prefer one measure over the other.

6.2 Results

As seen in Figure 5 and 6, the NE-based strategies outperform the word-based ones. Even the worst-performing NE-based measure (raw count, NEC) is better than the word-based strategies. TF-IDF, which is the most frequently mentioned strategy in the literature, outperforms raw counts. We can

[12] In these figures we show only positive values.

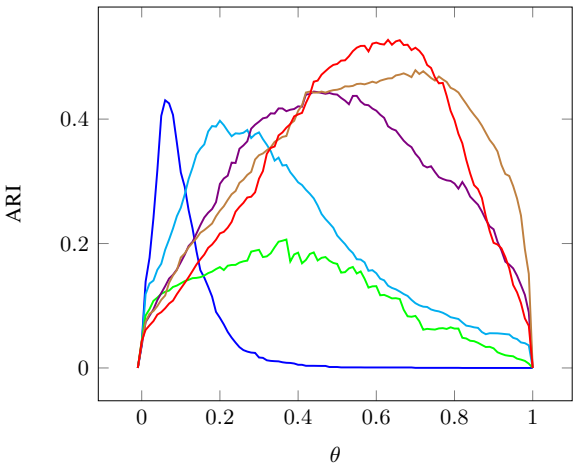

Figure 4: Adjusted Rand index

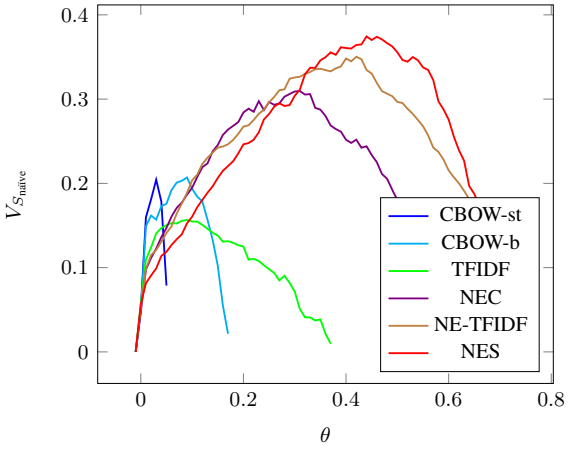

Figure 6: V Measure adjusted for $S_{\text{naïve}}$

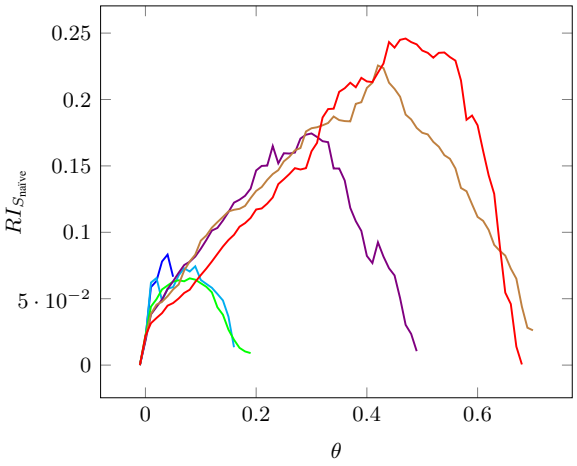

Figure 5: Rand index adjusted for $S_{\text{naïve}}$

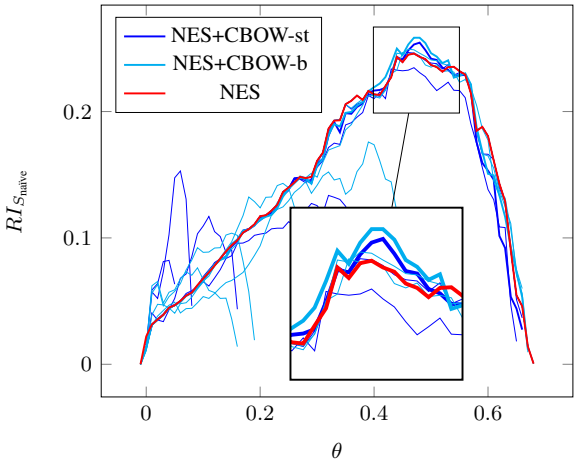

Figure 7: Rand index adjusted for $S_{\text{naïve}}$

argue for and against TF-IDF. For example, it is clear that locations that rarely pop up in news are more informative than popular country names. On the other hand, big companies, such as Google, are involved in many different activities and often appear in the news, which should not affect their relevance in a particular event. The best-performing measure, which is based on *salience* (NES), completely ignores the overall distribution of NEs in the corpus. However, it takes into account the *position* of the entity mentions in the text, and manages to outperform both raw counts and TF-IDF.

Among word-based measures, embeddings—CBOW-st and CBOW-b—outperform TF-IDF, and pre-trained embeddings, CBOW-st, are slightly better than the ones trained on our small business corpus, CBOW-b. It is also interesting how concave the CBOW plots are, as can be seen in Figures 5 and 6; this shows that the embedding representation has a clear, well-defined, threshold.

Figures 7 and 8 show the measures obtained by combining embedding-based representations and salience, according to Equation 1—juxtaposing combination method. We tested several values of α, on a logarithmic scale from $\frac{1}{1000}$ to 1000, some of them are shown using thin blue lines, dark for CBOW-st and light for CBOW-b. The thick lines present the best performing combinations, which correspond to $\alpha = 1$ for CBOW-st and $\alpha = 0.5$ for CBOW-b; the red curves present the values obtained by NES in figures 5 and 6.

It can be seen, from Figures 7 and 8, that, when combined with salience, the embeddings trained on our small domain-specific corpus outperform those trained on a much larger general corpus, even though CBOW-b alone performs worse than CBOW-st! It is interesting that, in the case of business-specific embeddings, it is better to give less weight to the NE features: the best α for CBOW-st is half of the best α for CBOW-b.

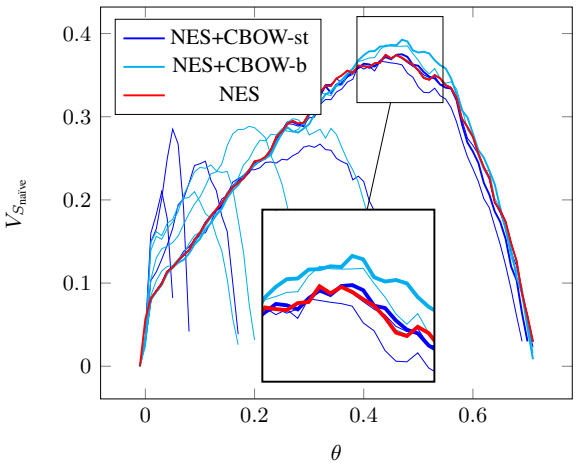

Figure 8: V-Measure adjusted for $S_{\text{naïve}}$

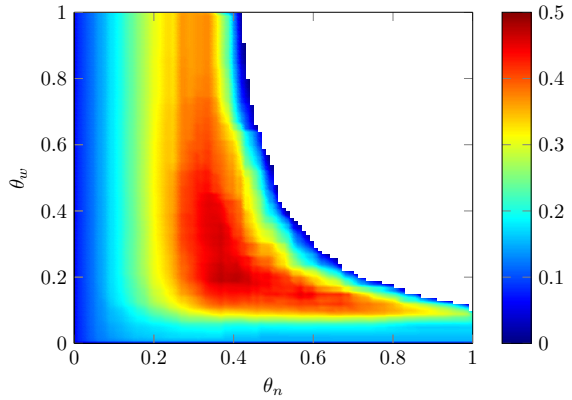

Figure 9: V-Measure adjusted for $S_{\text{naïve}}$

However, as can be seen from these figures, juxtaposed representations does not yield significant improvements over simply using a single set of features. Figures 9 and 10 show, respectively, the V-measure and Rand-index for CBOW-b and NES combined using the *AND* function: two documents are in the same group if and only if both distances are below their corresponding thresholds—θ_n for NEs and θ_w for words. Since we are optimizing these two parameters, we plot the results in a heat map. It can be seen that this approach to combination yields a significant improvement: up to 0.39 for $S_{\text{naïve}}$-adjusted Rand index and 0.47 for V-measure.

This significant improvement can be explained by the distribution of our corpus, presented in Figure 1, and by the fact that we use two representations of documents with different information. By using the *AND* function to combine them, we can filter out the cases where using only one representation would result in a false positive. In other

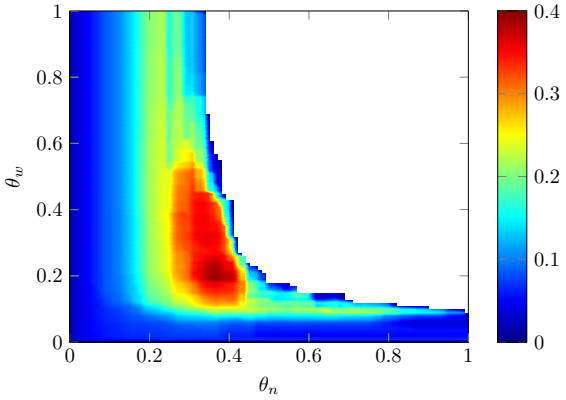

Figure 10: Rand index adjusted for $S_{\text{naïve}}$

words, two documents are grouped only if both their names and their common keywords are similar. We hypothesize that this can be a reasonable method for event-based clustering of news streams: the trending events are reported in many sources, while each source tries to produce some unique content. On the other hand, this may be not an appropriate strategy for topic classification.

7 Conclusions and future work

We have shown how considering the relative importance of named entities, in the form of *salience*, can be used to improve detection of related stories in different news articles. We have introduced an effective adjustment for the clustering metrics, and a method for combining different document vector representations, which outperforms the base representations alone. We make public the annotation interface, the news data, and the word embeddings used in this work, on our project page.[13]

We plan to explore other, more *user-oriented* metrics, which could take into account what a potential user might expect from a news-aggregating system. Other representations using the relative importance of named entities in a given news article should also be considered, such as a continuous-vector representation for documents where named entities play a role. Zhao and Karypis (2002) claim that agglomerative clustering may not be the best algorithm for this kind of task; therefore we plan to explore how the representations behave under other clustering algorithms.

[13] http://puls.cs.helsinki.fi/grouping

References

[Aggarwal and Zhai2012] Charu C Aggarwal and ChengXiang Zhai. 2012. A survey of text clustering algorithms. In *Mining text data*, pages 77–128. Springer.

[Azzopardi and Staff2012] Joel Azzopardi and Christopher Staff. 2012. Incremental clustering of news reports. *Algorithms*, 5(3):364–378.

[Berkhin2006] Pavel Berkhin. 2006. A survey of clustering data mining techniques. In *Grouping multidimensional data*, pages 25–71. Springer.

[Carpineto et al.2009] Claudio Carpineto, Stanislaw Osiński, Giovanni Romano, and Dawid Weiss. 2009. A survey of web clustering engines. *ACM Computing Surveys (CSUR)*, 41(3):17.

[Cheng et al.2012] Jia Cheng, Jingyu Zhou, and Shuang Qiu. 2012. Fine-grained topic detection in news search results. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, pages 912–917. ACM.

[Cutting et al.1992] Douglass R Cutting, David R Karger, Jan O Pedersen, and John W Tukey. 1992. Scatter/gather: A cluster-based approach to browsing large document collections. In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 318–329. ACM.

[Du et al.2016] Mian Du, Lidia Pivovarova, and Roman Yangarber. 2016. PULS: natural language processing for business intelligence. In *Proceedings of the 2016 Workshop on Human Language Technology*, pages 1–8. Go to Print Publisher.

[Gialampoukidis et al.2016] Ilias Gialampoukidis, Stefanos Vrochidis, and Ioannis Kompatsiaris. 2016. A hybrid framework for news clustering based on the dbscan-martingale and lda. In *Machine Learning and Data Mining in Pattern Recognition*, pages 170–184. Springer.

[Hubert and Arabie1985] Lawrence Hubert and Phipps Arabie. 1985. Comparing partitions. *Journal of classification*, 2(1):193–218.

[Huttunen et al.2012] Silja Huttunen, Arto Vihavainen, Mian Du, and Roman Yangarber. 2012. Predicting relevance of event extraction for the end user. In Thierry Poibeau, Horacio Saggion, Jakub Piskorski, and Roman Yangarber, editors, *Multi-source, Multilingual Information Extraction and Summarization*, Theory and Applications of Natural Language Processing, pages 163–176. Springer Berlin.

[Iglesias et al.2016] José Antonio Iglesias, Alexandra Tiemblo, Agapito Ledezma, and Araceli Sanchis. 2016. Web news mining in an evolving framework. *Information Fusion*, 28:90–98.

[Ji and Grishman2008] Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *ACL*, pages 254–262.

[Kiritoshi and Qiang2016] Keisuke Kiritoshi and M A Qiang. 2016. Named entity oriented difference analysis of news articles and its application. *IEICE TRANSACTIONS on Information and Systems*, 99(4):906–917.

[Kumaran and Allan2004] Giridhar Kumaran and James Allan. 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.

[Mikolov et al.2013a] Tomas Mikolov, Kai Chen, Greg S Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *NIPS*.

[Mikolov et al.2013b] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

[Montalvo et al.2012] Soto Montalvo, Víctor Fresno, and Raquel Martínez. 2012. Nesm: a named entity based proximity measure for multilingual news clustering. *Procesamiento del lenguaje natural*, 48:81–88.

[Pivovarova et al.2013] Lidia Pivovarova, Silja Huttunen, and Roman Yangarber. 2013. Event representation across genre. In *Proceedins of the* 1^{st} *Workshop on Events: Definition, Detection, Coreference, and Representation*, NAACL HLT.

[Rand1971] William M Rand. 1971. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850.

[Rayson and Garside2000] Paul Rayson and Roger Garside. 2000. Comparing corpora using frequency profiling. In *Proceedings of the workshop on Comparing Corpora*, pages 1–6. Association for Computational Linguistics.

[Rosenberg and Hirschberg2007] Andrew Rosenberg and Julia Hirschberg. 2007. V-measure: A conditional entropy-based external cluster evaluation measure. In *EMNLP-CoNLL*, volume 7, pages 410–420.

[Sankaranarayanan et al.2009] Jagan Sankaranarayanan, Hanan Samet, Benjamin E Teitler, Michael D Lieberman, and Jon Sperling. 2009. Twitterstand: news in tweets. In *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*, pages 42–51. ACM.

[Shi and Malik2000] Jianbo Shi and Jitendra Malik. 2000. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905.

[Staff et al.2015] Chris Staff, Joel Azzopardi, Colin Layfield, and Daniel Mercieca. 2015. Search results clustering without external resources. In *2015 26th International Workshop on Database and Expert Systems Applications (DEXA)*, pages 276–280. IEEE.

[Steinbach et al.2000] Michael Steinbach, George Karypis, Vipin Kumar, et al. 2000. A comparison of document clustering techniques. In *KDD workshop on text mining*, pages 525–526. Boston.

[Steinberger and Pouliquen2008] Ralf Steinberger and Bruno Pouliquen. 2008. Newsexplorer–combining various text analysis tools to allow multilingual news linking and exploration. *Lecture notes for the lecture held at the SORIA Summer School Cursos de Tecnologıas Lingüısticas*.

[Toda and Kataoka2005] Hiroyuki Toda and Ryoji Kataoka. 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.

[Vadrevu et al.2011] Srinivas Vadrevu, Choon Hui Teo, Suju Rajan, Kunal Punera, Byron Dom, Alexander J Smola, Yi Chang, and Zhaohui Zheng. 2011. Scalable clustering of news search results. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 675–684. ACM.

[van Erp et al.2014] Marieke van Erp, Gleb Satyukov, Piek Vossen, and Marit Nijsen. 2014. Discovering and visualising stories in news. In *LREC*, pages 3277–3282.

[Xu et al.2003] Wei Xu, Xin Liu, and Yihong Gong. 2003. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 267–273. ACM.

[Xu et al.2015] Jiaming Xu, Peng Wang, Guanhua Tian, Bo Xu, Jun Zhao, Fangyuan Wang, and Hongwei Hao. 2015. Short text clustering via convolutional neural networks. In *Proceedings of NAACL-HLT*, pages 62–69.

[Yangarber and Steinberger2009] Roman Yangarber and Ralf Steinberger. 2009. Automatic epidemiological surveillance from on-line news in MedISys and PULS. In *Proceedings of IMED-2009: International Meeting on Emerging Diseases and Surveillance*, Vienna, Austria.

[Yangarber2003] Roman Yangarber. 2003. Counter-training in discovery of semantic patterns. In *Proceedings of the* 41^{st} *Annual Meeting of the Association for Computational Linguistics*, Sapporo, Japan.

[Yangarber2006] Roman Yangarber. 2006. Verification of facts across document boundaries. In *Proceedings of the International Workshop on Intelligent Information Access (IIIA-2006)*, Helsinki, Finland.

[Zamir and Etzioni1999] Oren Zamir and Oren Etzioni. 1999. Grouper: a dynamic clustering interface to web search results. *Computer Networks*, 31(11):1361–1374.

[Zhang et al.2013] Jiwei Zhang, Qiuyue Dang, Yueming Lu, and Songlin Sun. 2013. Suffix tree clustering with named entity recognition. In *Cloud Computing and Big Data (CloudCom-Asia), 2013 International Conference on*, pages 549–556. IEEE.

[Zhao and Karypis2002] Ying Zhao and George Karypis. 2002. Evaluation of hierarchical clustering algorithms for document datasets. In *Proceedings of the eleventh international conference on Information and knowledge management*, pages 515–524. ACM.