Relational Summarization for Corpus Analysis

Abram Handler and **Brendan O'Connor**
College of Information and Computer Sciences
University of Massachusetts Amherst
ahandler@cs.umass.edu, brenocon@cs.umass.edu

Abstract

This work introduces a new problem, relational summarization, in which the goal is to generate a natural language summary of the relationship between two lexical items in a corpus, without reference to a knowledge base. Motivated by the needs of novel user interfaces, we define the task and give examples of its application. We also present a new query-focused method for finding natural language sentences which express relationships. Our method allows for summarization of more than two times more query pairs than baseline relation extractors, while returning measurably more readable output. Finally, to help guide future work, we analyze the challenges of relational summarization using both a news and a social media corpus.

1 Introduction

Research on automatic summarization (Nenkova et al., 2011; Das and Martins, 2007) aims to help users understand large document sets. However, the details of how textual summaries might actually be presented to users are often ignored. We propose that user interfaces which display noteworthy terms or concepts present the need for **relational summaries**: descriptions of the relationship between two entities or noun phrases from a corpus.

Examples of such interfaces include: commandline software for examining noteworthy terms or phrases (Squirrell, 2017; Robinson, 2016; Monroe et al., 2008), point-and-click browsers which display named entities and their interconnections on a network diagram (Wright et al., 2009; Görg et al., 2014; Tannier, 2016), concept map browsers (Falke and Gurevych, 2017b) and document search engines which suggest terms relevant to a query, such as the related searches displayed on Wikipedia info boxes from Google. In all such settings a natural question arises: what is the nature of the relationship between the entities or concepts shown in the interface? One particular interface which presents the need for a relational summary is shown in figure 1.

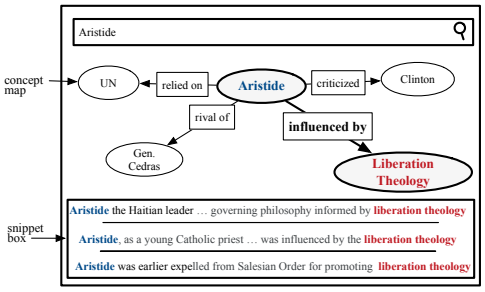

Figure 1: An example interface which requires relational summarization. The user has queried for the entity *Aristide*. The interface shows a *concept map* (top), displaying short summaries of Aristide's important relationships. The user has drilled down to see a more detailed summary of Aristide's relationship with liberation theology, displayed in a *snippet box* (bottom).

Relational questions are ubiquitous and varied. Examples include the following. What is the relationship between the "City of London" and "goal-delivery of Newgate" in 18th century court records (Hitchcock et al., 2012)? What is the relationship between "Advanced Integrated Systems" and "United Arab Emirates" in the Paradise Papers?[1] What does "dad" have to do with "mom" on the subreddit discussion forum *Relationship_Advice*?

This study seeks to answer such questions by examining the problem of **relational summarization**, which lies at the intersection of prior work on summarization and relation extraction. Unlike previous efforts at summarizing relationships (Falke and Gurevych, 2017a), our approach focuses on answering user queries about the connections between two particular terms, without ref-

[1] https://www.icij.org/investigations/paradise-papers/

Proceedings of NAACL-HLT 2018, pages 1760–1769
New Orleans, Louisiana, June 1 - 6, 2018.

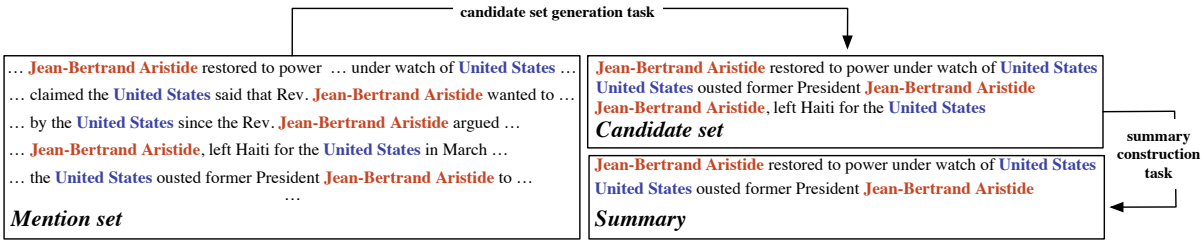

Figure 2: A relational summary is a synopsis of all sentences which mention two terms, denoted (t_1) and (t_2). We refer to such sentences as a *mention set*. In the figure above (t_1) is **Jean-Bertrand Aristide** and (t_2) is **United States**. To create a summary first requires identifying all statements in the mention set which coherently describe some relationship between (t_1) and (t_2). This **candidate set generation task** is a prerequisite for the subsequent **summary construction task**: selecting the top K candidates to create a summary. In this work, we offer a method for the first task and show how the second task will likely require a diversity of summarization techniques (§6).

erencing a knowledge graph (Voskarides et al., 2015).[2] In order to answer such queries we:

- Formally define the problem (§2), which we divide into two subtasks: **candidate set generation** and **summary construction**.
- Provide a new method for the candidate set generation task (§4), which we show outperforms baseline relation extraction techniques (§5) in terms of readability and yield.
- Analyze the summary construction task for future work (§6), demonstrating that different summarization techniques are likely most appropriate for different mention sets.

2 Formal definition and method

We refer to all sentences within a collection of documents which contain two terms, (t_1) and (t_2) as the *mention set*. (t_1) and (t_2) are noun phrases, a syntactic category which encompasses both traditional named entities like people and places, as well as less concrete, but important, entities and concepts like "liberation theology" (Handler et al., 2016).

A relational summary is a synopsis of the mention set. A summary consists of K relation statements, each displayed on its own line. Relation statements are natural language expressions which begin with (t_1) and end with (t_2). We refer to the span of tokens in between (t_1) and (t_2) as a *relation phrase*. We use the notation (t_1) r (t_2) to denote a relation statement, indicating two terms and a relation phrase. In the relation statement, "**Aristide** fled **Haiti**", r is the token "fled", (t_1) is the token **Aristide**, and (t_2) is the token **Haiti**.

Relation statements, which are strings intended for human readers, are similar to the 3-tuples, "*relations*", from prior work on information extraction (Banko et al., 2007). However, in this work, we show that the assumptions underlying the extraction of 3-tuples for machines (§3) leads to poor performance in summarizing mention sets for people (§5).

In this study, we present a strictly extractive method for generating relation statements: each relation statement must be constructed by deleting tokens from some sentence in the mention set.[3] Some relation statements constructed by deleting tokens from a sentence make sense; others do not. We refer to any (t_1) r (t_2) which makes makes sense to a human reader as **acceptable**.[4] Table 1 shows examples of acceptable and unacceptable relation statements, constructed by deletion.

s_1	Aristide fled Haiti ~~in 2004~~. (t_1) r (t_2)
s_2	~~For instance~~ Bush told Aristide ~~to leave~~. (t_1) r (t_2)

Table 1: Two relation statements constructed by deleting tokens from source sentences, s_1 and s_2. The relation statement extracted from s_1 is acceptable; the statement extracted from s_2 is not.

[2] Relational summaries are intended for general-purpose corpus analysis. Existing knowledge bases do not cover topics discussed in many corpora, such as historical court records (Hitchcock et al., 2012). Therefore, our approach does not employ a knowledge base.

[3] In subsequent studies of relation extractors (§5), we allow extractors to lightly introduce new tokens, such as adding the word "is" in relations expressed as noun phrases.

[4] Linguists sometimes use the term "acceptability" to refer to human judgements of the well-formedness of utterance. See Sprouse and Schütze (2014) for an overview.

Only acceptable relation statements are permitted in a summary. The set of all possible acceptable relation statements is called the **candidate set**, denoted $\mathcal{C}$. We refer to the task of identifying all acceptable relation statements as the **candidate set generation task**. Identifying a candidate set presents a subsequent problem of choosing the best collection of K relation statements from $\mathcal{C}$ to create a summary. We refer to this second step as the **summary construction task**.

As in traditional summarization (Das and Martins, 2007; Nenkova et al., 2011), a good relational summary should (i) be readable, (ii) include the most important aspects of the relationship between $(\boldsymbol{t_1})$ and $(\boldsymbol{t_2})$, (iii) avoid redundancy, and (iv) cover the full diversity of topics in the mention set.

Relational summaries might be presented with different kinds of user interfaces. In cases where a user seeks to browse many relationships, a summary might be displayed as a *concept map* (Falke and Gurevych, 2017a), where the two terms are vertexes in a directed graph and their relationship is printed along the edge label between them. In cases where user wants to investigate a specific relationship, a relational summary might be displayed as a *snippet box*: a short list of sentences which begin and end with the two terms. Figure 1 shows a snippet box and concept map. In a snippet box, both the number of lines in the summary and the length of the lines in the summary is longer than in a concept map.

3 Related work

Relational summarization intersects with a diversity of prior work from natural language processing, including work on **relation extraction**, **summarization** and **sentence compression**.

Traditionally, the goal of **relation extraction** is to cull structured facts for knowledge databases from unstructured text. Often, such facts take the form of a 3-tuple which defines a relationship between two arguments, such as (arg1=Angela Merkel, rel=met with, arg2=Theresa May). If extractors do not make use of a predefined schema, the task of finding relations is called Open Information Extraction (OpenIE). OpenIE systems[5] offer an off-the-shelf method for generating a candidate set for a relational summary. Their output can easily be linearized to $(\boldsymbol{t_1})\ r\ (\boldsymbol{t_2})$ statements by simply concatenating the three arguments of the triple to form a string.

However, we find that the recall of relation extractors is often too low to summarize many mention sets. We measure this disadvantage extensively in section §5.1. One reason for their poor performance might be that extractors have goals and assumptions which are poorly suited to the relation summarization task. In relation extraction, the aim is to find relation strings that recur for many different entity pairs, which allows such systems to build knowledge databases. For instance, relation extraction might be used to build tables of world leaders who rel="met with" other world leaders in order to analyze international politics. From this perspective, long, sparse, heterogenous and detailed relation strings which might apply only to a pair of specific arguments are undesirable, as they make it difficult to find general patterns across many different entity pairs. For example, the influential ReVerb OpenIE system (Fader et al., 2011) excludes "overly-specific relation phrases" which apply only to two entities. One way to help ensure that relations generalize across entity pairs is to strive for arguments which are as short as possible, a common goal in OpenIE (Stanovsky and Dagan, 2016).[6]

Our method for generating a candidate set is closer to approaches from **sentence compression** (Knight and Marcu, 2002; Clarke and Lapata, 2008; Filippova and Altun, 2013; Filippova et al., 2015), an NLP task which seeks to make a source sentence shorter while preserving the most important information and producing readable output. We show that our sentence compression approach allows us to achieve higher readability than off-the-shelf relation extractors (§5).

Sentence compression is often used in traditional extractive **summarization** to make more efficient use of a budgeted summary length. We discuss summarization further in §6, where we consider how existing work might be applied to the problem of selecting K statements from the candidate set.

[5]There are many available OpenIE systems. See Stanovsky and Dagan (2016) for an inventory of major work.

[6]Methods from the relation extraction literature which seek to deduce facts from extracted relations, such as Riedel et al. (2013), might also help identify useful summaries in future work. Relations which imply that other relations are true might make good summaries.

Sampled unacceptable compression	**Auburn police** are investigating the death of a **Tuskegee** ~~woman who died ...~~
Known acceptable compression	~~Drug firm~~ **Glenmark** has opened its new facility in **Argentina** ~~which would ...~~

Table 2: Examples of known acceptable and presumed unacceptable training examples, with entities shown in bold. We refer to crossed out spans as *outside of the compression*. Our model uses grammatical information from inside and outside of the compression to predict the acceptability of a compression.

4 Query-focused candidate set generation

Traditionally, relation extraction begins with a fixed notion of what constitutes a desirable "relation" between two arguments, defined by a predefined schema, a syntactic template (Fader et al., 2011), or a collection of seed examples (Angeli et al., 2015). The relation extraction task is then to correctly identify spans in which arguments are joined by a relation.

The relational summarization problem is somewhat different: we begin with a pair of query terms, $(\boldsymbol{t_1})$ and $(\boldsymbol{t_2})$, and we wish to learn the nature of their relationship. Therefore, any statement which coherently describes any relationship between the two query terms is potentially of interest, even if it does not match prior expectations of what constitutes a relation.

We thus approach the candidate set generation task as a specialized form of sentence compression: we attempt to predict if a sentence from the text can be coherently compressed to the form $(\boldsymbol{t_1})\ r\ (\boldsymbol{t_2})$. Table 2 shows examples of sentences which can and cannot be shortened to this form.

We use gold standard sentence–compression pairs from the Filippova and Altun (2013) dataset to supervise this prediction. In sentence compression corpora, gold standard compressions must be acceptable sentences. Therefore, compressions from the dataset which happen to begin and end with a named entity,[7] once extracted from source sentences, can serve as positive examples of acceptable relation statements. On the other hand, randomly chosen spans of the form $(\boldsymbol{t_1})\ r\ (\boldsymbol{t_2})$, which happen to arise in source sentences, are very often not acceptable as standalone sentences. These randomly sampled spans can serve as examples of unacceptable relation statements. We then predict acceptability with supervision from known gold acceptable and sampled, presumed incoherent examples.[8]

Filtering the original dataset in this manner[9] yields 17,529 positive and 30,266 negative sentences. We then downsample negative training examples to create two balanced classes of equal size, and use 81% of data for training, 9% for validation and the remaining 10% for testing.

Let $p(c = 1 \mid s, (\boldsymbol{t_1})\ r\ (\boldsymbol{t_2}))$ indicate the probability that a span of form $(\boldsymbol{t_1})\ r\ (\boldsymbol{t_2})$ extracted from sentence s is coherent. We model $p(c = 1|s, (\boldsymbol{t_1})\ r\ (\boldsymbol{t_2}))$ using logistic regression, with features based on the position of part-of-speech tags and dependency edges in s. Specifically, each sentence in the filtered dataset contains a span of the form $(\boldsymbol{t_1})\ r\ (\boldsymbol{t_2})$. We refer to the tokens in this span as *in the compression* because a user would see these tokens in a relation statement compressed from s. Each sentence also contains spans of tokens which are *outside of the compression* because they are deleted from the original source sentence to create a relation statement. Table 2 displays examples.

Our feature vector records the counts of how many times each part-of-speech tag in the tagset occurs in the compression and also independently records the counts of how many times each part-of-speech tag occurs out of the compression. We refer to the count of each part-of-speech tag in the compression and the count of each part-of-speech tag out of the compression as $\boldsymbol{\Phi}$. We also count the occurrence of each possible dependency edge label in the compression, and the count of each possible dependency edge label out of the compression. If a label's dependent lies in the compres-

[7] https://github.com/google-research-datasets/sentence-compression

[8] We manually inspect 100 negative examples, selected at random, and find that roughly 80% are in fact incoherent.

[9] We also exclude randomly chosen spans which happen to encompass the entire source sentence and exclude randomly chosen spans where $(\boldsymbol{t_1})$ and $(\boldsymbol{t_2})$ are joined by only edges of type compound in the dependency graph of the compression (e.g. "Coup leader Cedras ..."). We use CoreNLP version 3.8 to extract *enhanced++* Universal Dependencies (Manning et al., 2014; Schuster and Manning, 2016; Nivre et al., 2016). We also filter positive and negative examples where the span between $(\boldsymbol{t_1})$ and $(\boldsymbol{t_2})$ is longer than J=75 characters, to simulate a space constraint in a user interface. Finally, we remove all punctuation from the end of the sentence for both positive and negative examples because all gold positive compressions end in punctuation marks. For positive examples, if the compressed version of a sentence deletes tokens between t_1 and t_2, we replace the span between t_1 and t_2 in the source sentence with the compression.

$p(c = 1\|s, (\boldsymbol{t_1})\ r\ (\boldsymbol{t_2}))$	$(\boldsymbol{t_1})\ r\ (\boldsymbol{t_2})$
.005	**Jean-Bertrand Aristide** that the **United States**
.010	**United States** since the Rev. **Jean-Bertrand Aristide**
...	...
.894	**United States** ousted former President **Jean-Bertrand Aristide**
.976	**Jean-Bertrand Aristide**, left Haiti for the **United States**

Table 3: Highest and lowest coherence predictions from the set **United States – Jean-Bertrand Aristide**

sion, we consider the label in the compression.[10] We refer to these dependency edge counts as $\boldsymbol{\Psi}$. Our final feature vector, $\boldsymbol{\Omega}$, is defined as the concatenation of $\boldsymbol{\Psi}$ and $\boldsymbol{\Phi}$.

Features	Test accuracy
$\boldsymbol{\Phi}$ (pos)	.858
$\boldsymbol{\Psi}$ (deps)	.892
$\boldsymbol{\Omega}$ (deps & pos)	.896

Table 4: Test accuracies.

We implement our model with *scikit-learn* (Pedregosa et al., 2011) and manually tune the inverse regularization constant to the setting, $c = 1$, which achieves the highest accuracy on the validation set. For evaluation, a sentence is presumed coherent if $p(c = 1|s, (\boldsymbol{t_1})\ r\ (\boldsymbol{t_2})) > .5$. Using the feature vector $\boldsymbol{\Omega}$ we achieve an accuracy of .896 on the test set. We also present results using only the $\boldsymbol{\Psi}$ and $\boldsymbol{\Phi}$ features (table 4) because reliable dependency parses are not available in some settings (Blodgett et al., 2016; Bamman, 2017).

Two limitations of this approach suggest areas for future work. First, in some cases, the relationship between $(\boldsymbol{t_1})$ and $(\boldsymbol{t_2})$ might not be expressed in the form, $(\boldsymbol{t_1})\ r\ (\boldsymbol{t_2})$, as in "**Russia** and **France** signed an agreement". In order to generate candidate relation statements it would be helpful to lightly rewrite the sentence, as in "**Russia** signed an agreement with **France**". Additionally, a sentence might express a relationship between two terms but be too long to display on a concept map or a snippet box. In these cases, it would be helpful to compress the sentence to create a more concise relation statement.

5 Experiments

Any relational summarization system should deliver a high-quality summary when a user queries for two terms. Therefore, ideally, a system should generate the largest possible candidate set, without returning incoherent relation statements. We thus evaluate our query-focused generation method in terms of both readability and yield (total relation statements recalled). Our method generates three times more relation statements than OpenIE systems, allowing for summarization of two times more query pairs. We also achieve higher scores in a test of human coherence judgements (table 5).

More concretely, we evaluate our compression-based method for generating candidate sets against two relation extractor baselines on two very different corpora: (1) all comments from the large "relationships"[11] subreddit from June, 2015 – September, 2017[12] and (2) a collection of *New York Times* articles from 1987 to 2007 which mention the country "Haiti" (Sandhaus, 2008). For each corpus, we first find a collection of multi-word phrases using the `phrasemachine` package (Handler et al., 2016) which extracts all multi-word, noun phrase terms from the corpus.

After extracting all terms, we determine the top 100 terms, by count. We then examine all non-empty mention sets for all possible combinations of two top terms. A mention set is a set of sentences which mention two terms (§2). We examine all mention sets because an investigator should be able to investigate any entity she chooses while analyzing a corpus.

In subsequent experiments, we require all relation statements be less than or equal to $J = 75$ characters, which excludes overly verbose relation statements which are unsuitable for many user interfaces.

5.1 Yield experiments

Off-the-shelf relation extractors generate 3-tuples from each mention set. Some of those 3-tuples might have one argument which is equal to $(\boldsymbol{t_1})$ and another argument which is equal to $(\boldsymbol{t_2})$. Each such 3-tuple can be linearized into a string of the form $(\boldsymbol{t_1})\ r\ (\boldsymbol{t_2})$ to generate a candidate set. However, we find that using extractors in this

[10] Enhanced dependencies allow for a token to have more than one incoming edge (i.e., multiple parents). If there is more than one incoming edge, we pick an edge at random.

[11] "relationships" refers to interpersonal relationships

[12] https://medium.com/@jason_82699/pushshift-reddit-api-md-c2d70745c270

manner achieves a low yield (total number of extracted relations). A low yield is undesirable both because it limits the number of mention sets which may be summarized and generates fewer relation statements from which to select an optimal relational summary.

More precisely, we identify the 3-tuples which an OpenIE system extracts from a mention set such that exactly one argument from the triple is equal[13] to (t_1) and exactly one argument from the triple is equal to (t_2). We refer to these 3-tuples as "matching". We then count (1) the total number of mention sets which contain at least one matching 3-tuple and (2) the total number matching 3-tuples across all mention sets. We refer to such counts as the *yield* of a candidate generation system.

We measure the yield of Stanford OpenIE (Angeli et al., 2015) and ClausIE (Del Corro and Gemulla, 2013) on the *New York Times* and *Reddit* corpora, and compare each system to our compression-based approach (§4).[14] We measure these two relation extractors because Stanford OpenIE is included with the popular CoreNLP software and ClausIE achieves the highest recall in two systematic studies of relation extractors (Stanovsky and Dagan, 2016; Zhang et al., 2017).

We find that, for the great majority of sentences, relation extractors do not extract any relations between (t_1) and (t_2). Moreover, for many mention sets, the number of relations extracted with off-the-shelf systems is often zero. We show these results in table 5.

This suggests that although relation summarization is superficially similar to relation extraction, off-the-shelf extractors are poor tools for creating textual units to summarize mention sets. Very often, two terms are related to each other in ways which are simply not captured by relation extractors.

[13]Note that OpenIE systems might not extract the literal string (t_1) or (t_2) as arguments. For instance, if (t_1) is "Merkel" the OpenIE system might extract the argument "Angela Merkel". If some term and some argument from a relational triple share the same head token in the dependency parse of the sentence we say that they are equal. Falke and Gurevych (2017c) employ a similar equality criterion. We tokenize with CoreNLP. In extremely rare cases, tokenization mismatches between CoreNLP and ClausIE make it impossible to apply this criterion.

[14]For our compression-based approach, we count all cases where $p(c = 1 \mid s, (t_1)\ r\ (t_2)) > .5$ as extracting a relation statement.

5.2 Human acceptability judgments

Our compression-based method achieves higher yield than off-the-shelf relation extractors. However, because all sentences in a mention set include (t_1) and (t_2), it is always possible to generate a very large candidate set by simply extracting all spans between (t_1) and (t_2) from the mention set, regardless if such relation statements are coherent. We examine if gains in yield come at the expense of acceptability by presenting randomly selected relation statements to workers on the platform Figure Eight[15] (formerly Crowdflower) and asking workers to rate the extent to which they agree or disagree as to whether a relation statement is a "coherent English sentence" on a scale from 1 to 5. Each relation statement is shown to three workers in total.[16] Our approach is broadly similar to the readability experiments reported in Filippova and Altun (2013).

We solicit 481 total judgements from workers and calculate the mean acceptability score, by method and corpus (table 5). Our method achieves the highest mean acceptability score for both corpora.

Additionally, aggregating judgments across corpora, we observe a statistically significant (p=8 x 10^{-4}) difference between our method ($\mu = 3.89$, $\sigma = 1.38$) and Stanford OpenIE ($\mu = 3.33$, $\sigma = 1.46$) in a two-tailed t-test. Our method also achieves a higher mean score than ClausIE ($\mu = 3.69$, $\sigma = 1.44$), although the difference is not significant.

6 Future work: summary construction task

After a relational summarization system generates a candidate set, the next task is selecting the top K candidate statements for inclusion in a summary (figure 2). In this work, we do not attempt this summary construction task. However, in this section, we analyze the nature of the relational summarization challenge by describing differences among mention sets, and how these differences might affect future efforts at summarization.

We observe that mention sets are inherently heterogenous. Some describe a single, temporally-

[15]https://www.figure-eight.com/

[16]We use seven test questions to filter out careless or bad faith responses. Workers must answer 70% of test questions correctly to be included in a task's results. We construct test questions blindly, without knowledge of the system which generated the relation statement.

	Yield				Coherence	
	Total non-empty pairs		Total rel. stmts.		Mean judgment	
	Haiti	Reddit	Haiti	Reddit	Haiti	Reddit
ClauseIE	128	1,121	279	3,949	3.67	3.71
StanfordOIE	443	1,488	972	5,605	3.69	2.97
This work	**739**	**3,766**	**2,954**	**21,495**	**3.94**	**3.85**
Upper bound	2,472	4,496	43,051	123,760	Range: 1-5	

Table 5: We compare Stanford OpenIE, ClausIE and our headline-based compression method for the candidate set generation task on two different corpora (Haiti articles from *New York Times*, and the *Reddit* relationships forum) in terms of (1) how many entity pairs have a non-empty candidate set, (2) how many total relation statements are generated, and (3) the average human judgment of acceptability (§5.2). For yield measures, the upper bound on the left shows the total number of non-empty entity pairs (i.e. how many pairs actually cooccur in at least one sentence, out of all $\binom{100}{2} = 4950$ theoretically possible pairs) and the upper bound on the right shows the total number of sentences in the corpus which mention at least two of the terms. Our method summarizes more entity pairs across both corpora, and achieves the highest acceptability scores among all techniques (§5.2).

focused event. Others describe a consistent, unchanging relationship. Still others describe intricate sagas unfolding across time. For instance, within the Haiti corpus, one mention set describes events in 1994 when **General Cedras** fled to the **Dominican Republic**. This mention set is quite different from a set of sentences in the Reddit corpus in which users assert that **video games** are a **deal breaker** in interpersonal relationships. Figure 3 displays hand-crafted summarizes for these mention sets.

In general, the properties which guide how a mention set should be summarized are its **size**, **topical diversity**, **temporal focus** and the degree to which the set expresses **states or events**. In this section, we use the notation (t_1) – (t_2) to refer to a mention set. For instance, *New York – London* would refer to all sentences from a corpus which contain the names of both of these cities.

Size. In general, because many word types in a corpus occur infrequently (Zipf, 1949), the number of sentences which mention (t_1) and (t_2) is often small. For instance, of the 320,670 total sentences in the Haiti corpus, only 160 mention "Jean-Bertrand Aristide" and the "United States," which is nonetheless among the larger mention sets in the corpus. In general, larger sets often describe complex and noteworthy relationships, which are more difficult to summarize (figure 3c). Note that although individual mention sets are often small enough to simply read (unlike in some multi-document summarization settings), summarization of mention sets is still quite useful, as practitioners will often seek to understand many different relationships as they investigate a new topic (e.g. figure 1).

Topical diversity. In general, some mention sets are focused on a single topic, others are more diffuse. For instance, after losing power in a second, 2004 coup Haiti's Jean Bertrand Aristide was forced into exile in South Africa. The mention set for *Jean Bertrand Aristide – South Africa* contains twelve sentences which (mostly, but not exclusively) describe Aristide's removal from power and life in exile in South Africa from 2004 onwards. Detecting and including diverse or complex topics is a classic aim of traditional multi document summarization (e.g. Lin and Hovy (2000)), which might be applied in this new setting.

Temporal focus. In timestamped corpora such as news archives or social media posts, some mention sets are focused within certain time periods; others are spread across the span of the corpus. For instance, in the Haiti corpus, *General Cedras – Dominican Republic* are only mentioned together during a few months of 1994 (figure 3b). A good summary for this mention set should describe a central event from this time period: when General Cedras fled to the Dominican Republic. On the other hand, *Jean-Bertrand Aristide – United States* are mentioned together in 67 months in the corpus, covering a number of important events spread across decades (figure 3c). For this mention set, a narrow summary focusing on a single event would be inappropriate.

Many existing methods specialize in detecting (Chaney et al., 2016), tracking (Allan et al., 1998) and summarizing evolving topics in timestamped documents. Some systems focus specifically on summarizing event "spikes": both in news (e.g.

video games and I don't want that to be a **deal breaker**
video games was a **deal breaker**
video games is a **deal breaker**

(a) A hand-crafted summary for the mention set **video games–deal breaker**. The mention set contains many stative descriptions of the relationships between the two terms, indicating that a summary might focus on presenting fixed relationships rather than evolving events.

General Cedras ... last week fled to the **Dominican Republic**
Dominican Republic ... has indicated it will not permit permanent residence by **General Cedras**

(b) A hand-crafted summary for the mention set **General Cedras–Dominican Republic**. The set has a high number of mentions which all fall within a several month span, hinting at a relationship focused on a particular event at a particular point in time.

Aug. 1994 **United States** supports the restoration of the democratically elected president of Haiti, **Jean-Bertrand Aristide**
Oct. 1995 **Jean-Bertrand Aristide** was restored to power a year ago under the watch of **United States**
Sep. 2002 **United States** and other donors withheld contributions, hoping to spur President **Jean-Bertrand Aristide**
Mar. 2004 **Jean-Bertrand Aristide** asserted that he had been driven from power by the **United States**

(c) A hand-crafted summary for the mention set **Jean-Bertrand Aristide–United States**, one of the largest in the Haiti corpus. The mention set describes a complex, shifting relationship; at different times over several decades, Aristide was a beneficiary, opponent and critic of the United States.

Figure 3: Mention sets are heterogenous, requiring a diversity of summarization techniques. In this work, we analyze the diversity of mention sets towards future attempts that the relational summarization problem.

Alfonseca et al. (2013)) and on social media (e.g. Nichols et al. (2012)). In some cases, the event described in a mention set will even match the loose form of a common narrative template (Chambers and Jurafsky, 2008), such as when the two terms are codefendants at a trial.

Mention sets which are more temporally diffuse are also more challenging. Update summarization refers to summarizing changes introduced by new documents, possibly from a high volume stream (Kedzie et al., 2015). This form of summarization is important in cases when a relationship shifts or changes through time, as in figure 3c.

States or events. Mention sets may be coarsely divided into cases where the set expresses a stable state or property of the world in the eyes of the author (e.g. "England is a close ally of the US" or "video games are a deal breaker") and cases where the relation statement expresses a change or event (e.g. "Gen. Cedras fled to the Dominican Republic" or "dad left mom"). In many interesting cases, the mention set contains a mix of stative and eventive relation statements which express a narrative, such as "**America** is an ally of **South Korea**" and "**America** sent a destroyer to **South Korea**".

Defining (Pustejovsky, 1991), extracting (Aguilar et al., 2014) and determining relationships between events (Hovy et al., 2013) is a challenging research area. But a better understanding of states and events would improve future work. For instance, if a summary includes the event "Jolie divorced Pitt", it does not need to include the stative relation phrase "Jolie was married to Pitt". To our knowledge, there is no prior work which considers how fine-grained relations between states and events might be employed for summarization. MacCartney and Manning (2009) offer a framework which might serve as a useful starting catalog.

Conclusion

This work defines a problem which lies at the intersection of typically unrelated fields in natural language processing, summarization and relation extraction. We present a new method which finds large numbers of natural language expressions which coherently describe relationships. We also analyze the challenges of the relational summarization task, by investigating and describing the inherent heterogeneity of mention sets. Because of this heterogeneity, we argue that future attempts to summarize relationships will likely require a diversity of models and techniques.

Acknowledgments

Thanks to Emma Strubell, Patrick Verga, Haw-Shiuan Chang, Su Lin Blodgett, Katherine Keith and the UMass NLP reading group for helpful discussions and comments. Thanks to Brian Dillon for helping us better understand how to collect and interpret human judgements of linguistic acceptability.

References

Jacqueline Aguilar, Charley Beller, Paul McNamee, Benjamin Van Durme, Stephanie Strassel, Zhiyi Song, and Joe Ellis. 2014. A comparison of the events and relations across ACE, ERE, TAC-KBP, and FrameNet annotation standards. In *Proceedings of the Second Workshop on EVENTS: Definition, Detection, Coreference, and Representation*. Association for Computational Linguistics.

Enrique Alfonseca, Daniele Pighin, and Guillermo Garrido. 2013. Heady: News headline abstraction through event pattern clustering. In *ACL*.

James Allan, Jaime Carbonell, George Doddington, Jonathan Yamron, and Yiming Yang. 1998. Topic detection and tracking pilot study: Final report.

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *ACL*.

David Bamman. 2017. Natural language processing for the long tail. *Digital Humanities* .

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *IJCAI*.

Su Lin Blodgett, Lisa Green, and Brendan T. O'Connor. 2016. Demographic dialectal variation in social media: A case study of african-american english. In *EMNLP*.

Nathanael Chambers and Dan Jurafsky. 2008. Unsupervised learning of narrative event chains. In *Proceedings of ACL-08: HLT*.

Allison Chaney, Hanna Wallach, Matthew Connelly, and David Blei. 2016. Detecting and characterizing events. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas.

James Clarke and Mirella Lapata. 2008. Global inference for sentence compression: An integer linear programming approach. *Journal of Artificial Intelligence Research* 31:399–429.

Dipanjan Das and André F. T. Martins. 2007. A survey on automatic text summarization. Technical report, Literature Survey for the Language and Statistics II course at Carnegie Mellon University.

Luciano Del Corro and Rainer Gemulla. 2013. Clausie: clause-based open information extraction. In *Proceedings of the 22nd international conference on World Wide Web*. ACM.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *EMNLP*. Edinburgh, Scotland, UK.

Tobias Falke and Iryna Gurevych. 2017a. Bringing structure into summaries: Crowdsourcing a benchmark corpus of concept maps. In *EMNLP*.

Tobias Falke and Iryna Gurevych. 2017b. Graphdocexplore: A framework for the experimental comparison of graph-based document exploration techniques. In *EMNLP: System Demonstrations*.

Tobias Falke and Iryna Gurevych. 2017c. Utilizing automatic predicate-argument analysis for concept map mining. In *Proceedings of the 12th International Conference on Computational Semantics (IWCS)*.

Katja Filippova, Enrique Alfonseca, Carlos A Colmenares, Lukasz Kaiser, and Oriol Vinyals. 2015. Sentence compression by deletion with LSTMs. In *EMNLP*.

Katja Filippova and Yasemin Altun. 2013. Overcoming the lack of parallel data in sentence compression. In *EMNLP*.

Carsten Görg, Zhicheng Liu, and John Stasko. 2014. Reflections on the evolution of the jigsaw visual analytics system. *Information Visualization* 13(4):336–345.

Abram Handler, Matthew J Denny, Hanna Wallach, and Brendan OConnor. 2016. Bag of what? Simple noun phrase extraction for text analysis. *Workshop on NLP + CSS, EMNLP* .

Tim Hitchcock, Robert Shoemaker, Clive Emsley, Sharon Howard, and Jamie McLaughlin. 2012. The old bailey proceedings online, 1674-1913. www.oldbaileyonline.org.

Eduard Hovy, Teruko Mitamura, Felisa Verdejo, Jun Araki, and Andrew Philpot. 2013. Events are not simple: Identity, non-identity, and quasi-identity. In *NAACL*.

Chris Kedzie, Kathleen McKeown, and Fernando Diaz. 2015. Predicting salient updates for disaster summarization. In *ACL*.

Kevin Knight and Daniel Marcu. 2002. Summarization beyond sentence extraction: A probabilistic approach to sentence compression. *Artificial Intelligence* 139(1):91–107.

Chin-Yew Lin and Eduard Hovy. 2000. The automated acquisition of topic signatures for text summarization. In *Proceedings of the 18th conference on Computational linguistics-Volume 1*. Association for Computational Linguistics.

Bill MacCartney and Christopher D Manning. 2009. An extended model of natural logic. In *Proceedings of the eighth international conference on computational semantics*.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Association for Computational Linguistics, Baltimore, Maryland, pages 55–60. http://www.aclweb.org/anthology/P14-5010.

Burt L Monroe, Michael P Colaresi, and Kevin M Quinn. 2008. Fightin' words: Lexical feature selection and evaluation for identifying the content of political conflict. *Political Analysis* 16(4):372–403.

Ani Nenkova, Kathleen McKeown, et al. 2011. Automatic summarization. *Foundations and Trends in Information Retrieval* 5(2–3):103–233.

Jeffrey Nichols, Jalal Mahmud, and Clemens Drews. 2012. Summarizing sporting events using twitter. In *IUI*.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal dependencies v1: A multilingual treebank collection. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. European Language Resources Association (ELRA), Paris, France.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.* 12:2825–2830.

James Pustejovsky. 1991. The syntax of event structure. *Cognition* 41(1):47–81.

Sebastian Riedel, Limin Yao, Benjamin M. Marlin, and Andrew McCallum. 2013. Relation extraction with matrix factorization and universal schemas. In *NAACL*.

David Robinson. 2016. Text analysis of trump's tweets confirms he writes only the (angrier) android half. http://varianceexplained.org/r/trump-tweets/.

Evan Sandhaus. 2008. The New York Times Annotated Corpus. *Linguistic Data Consortium* LDC2008T19.

Sebastian Schuster and Christopher D. Manning. 2016. Enhanced english universal dependencies: An improved representation for natural language understanding tasks. In *LREC*.

John Sprouse and Carson Schütze. 2014. *Research Methods in Linguistics*, Cambridge University Press, Cambridge, UK, chapter Judgment Data.

Tim Squirrell. 2017. Linguistic data analysis of 3 billion Reddit comments shows the alt-right is getting stronger. https://qz.com/1056319/what-is-the-alt-right.

Gabriel Stanovsky and Ido Dagan. 2016. Creating a large benchmark for open information extraction. In *EMNLP*. Austin, Texas.

Xavier Tannier. 2016. NLP-driven data journalism: Time-aware mining and visualization of international alliances. In *Proceedings of the 2016 IJCAI Workshop on Natural Language Processing meets Journalism*.

Nikos Voskarides, Edgar Meij, Manos Tsagkias, Maarten De Rijke, and Wouter Weerkamp. 2015. Learning to explain entity relationships in knowledge graphs. In *ACL*.

Brandon Wright, Jason Payne, Matthew Steckman, and Scott Stevson. 2009. Palantir: A visualization platform for real-world analysis. In *Visual Analytics Science and Technology, 2009. VAST 2009. IEEE Symposium on*. IEEE, pages 249–250.

Sheng Zhang, Rachel Rudinger, and Benjamin Van Durme. 2017. An Evaluation of PredPatt and Open IE via Stage 1 Semantic Role Labeling. In *The Proceedings of the 12th International Conference on Computational Semantics (IWCS)*.

George K. Zipf. 1949. *Human Behavior and the Principle of Least Effort*. Addison-Wesley.