Mining Paraphrasal Typed Templates from a Plain Text Corpus

Or Biran
Columbia University
orb@cs.columbia.edu

Terra Blevins
Columbia University
tlb2145@columbia.edu

Kathleen McKeown
Columbia University
kathy@cs.columbia.edu

Abstract

Finding paraphrases in text is an important task with implications for generation, summarization and question answering, among other applications. Of particular interest to those applications is the specific formulation of the task where the paraphrases are templated, which provides an easy way to lexicalize one message in multiple ways by simply plugging in the relevant entities. Previous work has focused on mining paraphrases from parallel and comparable corpora, or mining very short sub-sentence synonyms and paraphrases. In this paper we present an approach which combines distributional and KB-driven methods to allow robust mining of sentence-level paraphrasal templates, utilizing a rich type system for the slots, from a plain text corpus.

1 Introduction

One of the main difficulties in Natural Language Generation (NLG) is the *surface realization* of messages: transforming a message from its internal representation to a natural language phrase, sentence or larger structure expressing it. Often the simplest way to realize messages is though the use of templates. For example, any message about the birth year and place of any person can be expressed with the template "[Person] was born in [Place] in [Year]".

Templates have the advantage that the generation system does not have to deal with the internal syntax and coherence of each template, and can instead focus on document-level discourse coherence and on local coreference issues. On the other hand, templates have two major disadvantages. First, having a human manually compose a template for each possible message is costly, especially when a generation system is relatively open-ended or is expected to deal with many domains. In addition, a text generated using templates often lacks variation, which means the system's output will be repetitive, unlike natural text produced by a human.

In this paper, we are concerned with a task aimed at solving both problems: automatically mining paraphrasal templates, i.e. groups of templates which share the same slot types and which, if their slots are filled with the same entities, result in paraphrases. We introduce an unsupervised approach to paraphrasal template mining from the text of Wikipedia articles.

Most previous work on paraphrase detection focuses either on a corpus of aligned paraphrase candidates or on such candidates extracted from a parallel or comparable corpus. In contrast, we are concerned with a very large dataset of templates extracted from a single corpus, where any two templates are potential paraphrases. Specifically, paraphrasal templates can be extracted from sentences which are not in fact paraphrases; for example, the sentences "The population of Missouri includes more than 1 million African Americans" and "Roughly 185,000 Japanese Americans reside in Hawaii" can produce the templated paraphrases "The population of [american state] includes more than [number] [ethnic group]" and "Roughly [number] [ethnic group] reside in [american state]". Looking for paraphrases among templates, instead of among sentences, allows us to avoid using an aligned corpus.

Our approach consists of three stages. First, we process the entire corpus and determine slot locations, transforming the sentences to templates (Section 4). Next, we find most approriate *type* for each slot using a large taxonomy, and group together templates which share the same set of types

as potential paraphrases (Section 5). Finally, we cluster the templates in each group into sets of paraphrasal templates (Section 6).

We apply our approach to six corpora representing diverse subject domains, and show through a crowd-sourced evaluation that we can achieve a high precision of over 80% with a reasonable similarity threshold setting. We also show that our threshold parameter directly controls the trade-off between the number of paraphrases found and the precision, which makes it easy to adjust our approach to the needs of various applications.

2 Related Work

To our knowledge, although several works exist which utilize paraphrasal templates in some way, the task of extracting them has not been defined as such in the literature. The reason seems to be a difference in priorities. In the context of NLG, Angeli et al. (2010) as well as Kondadadi et al. (2013) used paraphrasal templates extracted from aligned corpora of text and data representations in specific domains, which were grouped by the data types they relate to. Duma and Klein (2013) extract templates from Wikipedia pages aligned with RDF information from DBPedia, and although they do not explicitly mention aligning multiple templates to the same set of RDF templates, the possibility seems to exist in their framework. In contrast, we are interested in extracting paraphrasal templates from non-aligned text for general NLG, as aligned corpora are difficult to obtain for most domains.

While template extraction has been a relatively small part of NLG research, it is very prominent in the field of Information Extraction (IE), beginning with Hearst (1992). There, however, the goal is to extract good data and not to extract templates that are good for generation. Many pattern extraction (as it is more commonly referred to in IE) approaches focus on semantic patterns that are not coherent lexically or syntactically, and the idea of paraphrasal templates is not important (Chambers and Jurafsky, 2011). One exception which expicitly contains a paraphrase detection component is (Sekine, 2006).

Meanwhile, independently of templates, detecting paraphrases is an important, difficult and well-researched problem of Natural Language Processing. It has implications for the general study of semantics as well as many specific applications such as Question Answering and Summarization. Research that focuses on mining paraphrases from large text corpora is especially relevant for our work. Typically, these approaches utilize a parallel (Barzilay and McKeown, 2001; Ibrahim et al., 2003; Pang et al., 2003; Quirk et al., 2004; Fujita et al., 2012; Regneri and Wang, 2012) or comparable corpus (Shinyama et al., 2002; Barzilay and Lee, 2003; Sekine, 2005; Shen et al., 2006; Zhao et al., 2009; Wang and Callison-Burch, 2011), and there have been approaches that leverage bilingual aligned corpora as well (Bannard and Callison-Burch, 2005; Madnani et al., 2008).

Of the above, two are particularly relevant. Barzilay and Lee (2003) produce *slotted lattices* that are in some ways similar to templates, and their work can be seen as the most closely related to ours. However, as they rely on a comparable corpus and produce untyped slots, it is not directly comparable. In our approach, it is precisely the fact that we use a rich type system that allows us to extract paraphrasal templates from sentences that are not, by themselves, paraphrases and avoid using a comparable corpus. Sekine (2005) produces typed phrase templates, but the approach does not allow learning non-trivial paraphrases (that is, paraphrases that do not share the exact same keywords) from sentences that do not share the same entities (thus remaining dependent on a comparable corpus), and the type system is not very rich. In addition, that approach is limited to learning short paraphrases of relations between two entities.

Another line of research is based on contextual similarity (Lin and Pantel, 2001; Paşca and Dienes, 2005; Bhagat and Ravichandran, 2008). Here, shorter (phrase-level) paraphrases are extracted from a single corpus when they appear in a similar lexical (and in later approaches, also syntactic) context. The main drawbacks of these methods are their inability to handle longer paraphrases and their tendency to find phrase pairs that are semantically related but not real paraphrases (e.g. antonyms or taxonomic siblings).

More recent work on paraphrase detection has, for the most part, focused on classifying provided sentence pairs as paraphrases or not, using the Microsoft Paraphrase Corpus (Dolan et al., 2004). Mihalcea et al. (2006) evaluated a wide range of lexical and semantic measures of similarity and introduced a combined metric that outperformed all previous measures. Madnani et al. (2012) showed that metrics from Machine Translation can be used

to find paraphrases with high accuracy. Another line of research uses the similarity of texts in a latent space created through matrix factorization (Guo and Diab, 2012; Ji and Eisenstein, 2013). Other approaches that have been explored are explicit alignment models (Das and Smith, 2009), distributional memory tensors (Baroni and Lenci, 2010) and syntax-aware representations of multi-word phrases using word embeddings (Socher et al., 2011). Word embeddings were also used by Milajevs et al. (2014). These approaches are not comparable to ours because they focus on classification, as opposed to mining, of paraphrases.

Detecting paraphrases is closely related to research on the mathematical representation of sentences and other short texts, which draws on a vast literature on semantics, including but not limited to lexical, distributional and knowledge-based semantics. Of particular interest to us is the work of Blacoe and Lapata (2012), which show that simple combination methods (e.g., vector multiplication) in classic vector space representations outperform more sophisticated alternatives which take into account syntax and which use deep representations (e.g. word embeddings, or the distributional memory approach). This finding is appealing since classic vector space representation (distributional vectors) are easy to obtain and are interpretable, making it possible to drill into errors.

3 Taxonomy

Our method relies on a type system which links entities to one another in a taxonomy. We use a combination of WordNet (Fellbaum, 1998) and DBPedia (Auer et al., 2007), which provides both a rich top-level type system with lexicalizations of multiple senses and a large database of entities linked through the type system (the top-level DBPedia categories all have cognates in WordNet, which make the two easy to combine). Leveraging the fact that DBPedia entities have corresponding Wikipedia pages, we also use the *redirect* terms for those pages as alternative lexicalizations of the entity (e.g., the Wikipedia article "United States" has "USA" as a redirect term, among others).

4 Creating Templates

The first step to creating the templates is to find entities, which are candidates to becoming slots in the templates. Since we are trying to find sentence-level paraphrasal templates, each sentence in the corpus is a potential template.

Entities are found in multiple ways. First, we use regular expressions to find dates, percentages, currencies, counters (e.g., "9th") and general numbers. Those special cases are immediately given their known type (e.g., "date" or "percentage"). Next, after POS-tagging the entire corpus, we look for candidate entities of the following kinds: terms that contain only NNP (including NNPS) tags; terms that begin and end with an NNP and contain only NNP, TO, IN and DT tags; and terms that contain only capitalized words, regardless of the POS tags. Of these candidates, we only keep ones that appear in the taxonomy. Unlike the special cases above, the type of the slots created from these general entities is not yet known and will be decided in the next step.

At the end of this step, we have a set of partially-typed templates: one made from each sentence in the corpus, with its slots (but not their types in most cases) defined by the location of entities. We remove from this set all templates which have less than two slots as these are not likely to be interesting, and all templates which have more than five slots to avoid excessively complicated templates.

We originally experimented with simply accepting any term that appears in the taxonomy as an entity. That method, however, resulted in a large number of both errors and entities that were too general to be useful (e.g, "table", "world" and similar terms are in the taxonomy). Note that NER approaches, even relatively fine-grained ones, would not give us the same richness of types that directly comparing to the taxonomy allows (and the next step requires that each entity we handle exist in the taxonomy, anyway).

5 Template Typing and Grouping

Determining the type of a slot in the template presents two difficulties. First, there is a sense disambiuation problem, as many lexical terms have more than one sense (that is, they can correspond to more than one entry in the taxonomy). Second, even if the sense is known, it is not clear which level of the taxonomy the type should be chosen from. For example, consider the sentence "[JFK] is [New York]'s largest airport" (the terms in square brackets will become slots once their types are determined). "JFK" is ambiguous: it can be an airport, a president, a school, etc. The first

step in this process is, then, to determine which of the possible senses of the term best fits the sentence. But once we determine that the sense of "JFK" here is of an airport, there are different types we can choose. JFK is a New York Airport, which is a type of Airport, which is a type of Air Field, which is a type of Facility and so on. The specificity of the type we choose will determine the correctness of the template, and also which other templates we can consider as potential paraphrases.

Our solution is a two-stage distributional approach: *choosing the sense*, and then *choosing the type level* that best fit the context of the slot. In each stage, we construct a $pseudo-sentence$ (a collection of words in arbitrary, non-grammatical order) from words used in the taxonomy to describe each option (a sense in the first stage, and a type level in the second stage), and then use their vector representations to find the option that best matches the context.

Following the observation of Blacoe and Lapata (2012) that simple similarity metrics in traditional vector representations match and even outperform more sophisticated representations in finding relations among short texts as long as multiplication is used in forming vector representations for the texts, we use traditional context vectors as the basis of our comparisons in both stages. We collect context vectors from the entire English Wikipedia corpus, with a token window of 5. To avoid noise from rarely occuring words and reduce the size of the vectors, we remove any feature with a count below a threshold of $log_{10}(\Sigma)$ where Σ is the sum of all feature counts in the vector. Finally, the vector features are weighted with (normalized) TF*IDF.[1]

For a multi-word collection (e.g. a pseudo-sentence) ψ, we define the features of the combined vector V_ψ using the vectors of member words V_w as:

$$V_{j\psi} = (\prod_{w \in \psi} V_{jw})^{\frac{1}{|S|}} \quad (1)$$

Where V_{jw} is the value of the jth feature of V_w.

To choose the sense of the slot (the first stage), we start with S, the set of all possible senses (in the taxonomy) for the entity in the slot. We create a pseudo-sentence ψ_s from the primary lexicalizations of all types in the hierarchy above each sense s - e.g., for the airport sense of JFK we create a single pseudo-sentence $\psi_{JFK-airport-sense}$ consisting of the terms "New York airport", "airport", "air field", "facility" and so on.[2] We create a vector representation V_{ψ_s} for each ψ_s using Equation 1. Then, we create a pseudo-sentence $\psi_{context}$ for the context of the slot, composed of the words in a 5-word window to the left and right of the slot in the original sentence, and create the vector $V_{\psi_{context}}$. We choose the sense $\hat{s}$ with the highest cosine similarity to the contex:

$$\hat{s} = \arg\max_{s \in S} \cos(V_{\psi_s}, V_{\psi_{context}})$$

Note that this is a deep similarity - the similarity of the (corpus) context of the sense and the (corpus) context of the slot context; the words in the sentence themselves are not used directly.

We use the lexicalizations of all types in the hierarchy to achieve a more robust vector representation that has higher values for features that co-occur with many levels in the sense's hierarchy. For example, we can imagine that "airplane" will co-occur with many of the types for the JFK airport sense, but "La Guardia" will not (helping to lower the score of the first, too-specific sense of "New York airport") and neither will features that co-occur with other senses of a particular type - e.g., "Apple" for the "airport" type.[3]

Once the sense is chosen, we choose the proper type level to use (the second stage). Here we create a pseudo-sentence for each type level separately, composed of all possible lexicalizations for the type. For example, the "air field" type contains the lexicalizations "air field", "landing field", and "flying field". These pseudo-sentences are then compared to the context in the same way as above, and the one with highest similarity is chosen. The reason for using all lexicalizations is similar to the one for using all types when determining the sense: to create a more robust representation that down-scores arbitrary co-occurences.

At the end of this step, the templates are fully typed. Before continuing to the next step of finding paraphrases, we group all *potential* paraphrases together. Potential paraphrases are simply

[1] A "term" being a single feature count, and a "document" being a vector

[2] But we exclude a fixed, small set of the most abstract types from the first few levels of the WordNet hierarchy, as these turn out to never be useful

[3] AirPort is the name of an Apple product

groups of templates which share exactly the same set of slot types (regardless of ordering).

6 Finding Paraphrases within Groups

Each group of potential paraphrases may contain multiple sub-groups such that each of the members of the subgroup is a paraphrase of all the others. In this last stage, we use a clustering algorithm to find these sub-groups.

We define the distance between any two templates in a group as the Euclidean distance between the vectors (created using Equation 1) of the two templates with the entity slots removed (that is, the pseudo-sentences created with all words in the template outside of the slots). We tried other distance metrics as well (for example, averaging the distances between the contexts surrounding each pair of corresponding slots in both templates) but the Euclidean distance seemed to work best.

Using this metric, we apply single-linkage agglomerative clustering, with the stopping criteria defined as a threshold τ for the maximum sum of squared errors (SSE) within any cluster. Specifically, the algorithm stops linking if the cluster C that would be created by the next link satisfies:

$$log(\sum_{v}^{C} d(v, \mu_C)^2) \geq \tau$$

Where μ_C is the centroid of C and d is the Euclidean distance. The logarithm is added for convenience, since the SSE can get quite large and we want to keep τ on a smaller scale.

The intuition behind this algorithm is that some paraphrases will be very similar (lexically or on a deeper level) and easy to find, while some will be more difficult to distinguish from template pairs that are related but not paraphrasal. The single-linkage approach is essentially transductive, allowing the most obvious clusters to emerge first and avoiding the creation of a central model that will become less precise over time. The threshold is a direct mechanism for controlling the trade-off between precision and recall.

At the end of this step, any pair of templates within the same cluster is considered a paraphrase. Clusters that contain only a single template are discarded (in groups that have high distances among their member templates, often the entire group is discarded since even a single link violates the threshold).

7 Evaluation

To evaluate our method, we applied it to the six domains described in Table 1. We tried to choose a set of domains that are diverse in topic, size and degree of repeated structure across documents. For each domain, we collected a corpus composed of relevant Wikipedia articles (as described in the table) and used the method described in Sections 4-6 to extract paraphrasal templates. We used Wikipedia for convenience, since it allows us to easily select domain corpora, but there is nothing in our approach that is specific to Wikipedia; it can be applied to any text corpus.

We sampled 400 pairs of paraphrases extracted from each domain and used this set of 2400 pairs to conduct a crowd-sourced human evaluation on CrowdFlower. For each template pair, we randomly selected one and used its original entities in both templates to create two sentences about the same set of entities. The annotators were presented with this pair and asked to score the extent to which they are paraphrases on a scale from 1 to 5. Table 2 shows the labels and a brief version of the explanations provided for each. To ensure the quality of annotations, we used a set of hidden test questions throughout the evaluation and rejected the contributions of annotators which did not get at least 70% of the test questions correctly. Of those that did perform well on the test questions, we had three annotators score each pair and used the average as the final score for the pair. In 39.4% of the cases, all three annotators agreed; two annotators agreed in another 47% of the cases, and in the remaining 13.6% there was complete disagreement. The inter-annotator agreement for the two annotators that had the highest overlap (27 annotated pairs), using Cohen's Kappa, was $\kappa = 0.35$.

The overall results are shown in Figure 1. Note that because of our clustering approach, we have a choice of similarity threshold. The results are shown across a range of thresholds from 8 to 11 - it is clear from the figure that the threshold provides a way to control the trade-off between the number of paraphrases generated and their precision. Table 3 shows the results with our preferred threshold of 9.5.

The number of paraphrase clusters found changes with the threshold. For the 9.5 threshold we find 512 clusters over all domains, a little over 60% of the number of paraphrases. The distribution of their sizes is Zipfian: a few very large clus-

Domain	Description	Size	Source article link
NBA	NBA teams	30	`National_Basketball_Association`
States	US states	50	N/A
AuMa	Automobile manufacturers	241	`List_of_automobile_manufacturers`
Metal	Heavy Metal bands (original movement, 1967-1981)	291	`List_of_heavy_metal_bands`
CWB	Battles of the American Civil War	446	`List_of_American_Civil_War_battles`
Marvel	Superheroes from the Marvel Comics universe	932	`Category:Marvel_Comics_superheroes`

Table 1: Evaluation domains. Source article links are preceded by https://en.wikipedia.org/wiki/

Score	Label	Explanation
5	Perfect Paraphrase	The two sentences are equivalent in meaning (but allow differences in e.g. tense, wordiness or sentiment)
4	Almost Paraphrase	The two sentences are equivalent in meaning with one minor difference (e.g., change or remove one word)
3	Somewhat Paraphrase	The two sentences are equivalent in meaning with a few minor differences, or are complex sentences with a part that is a paraphrase and a part that is not
2	Related	The sentences are related in meaning, but are not paraphrases
1	Unrelated	The meanings of the sentences are unrelated

Table 2: Annotation score labels and explanations

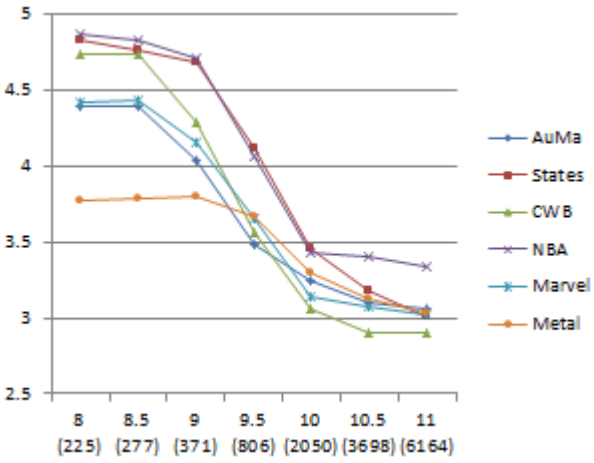

Figure 1: The average scores for each domain, for a range of threshold choices. The number in parentheses for each threshold is the number of paraphrases generated

Domain	# paraphrases	Avg.	%3+	%4+
NBA	30	4.1	88%	70%
States	171	4.1	86%	76%
AuMa	58	3.5	80%	50%
Metal	98	3.7	82%	63%
CWB	81	3.6	75%	56%
Marvel	428	3.7	83%	63%

Table 3: Size, average score, % of pairs with a score above 3 (paraphrases), and % of pairs with a score above 4 (high quality paraphrases) for the different domains with a 9.5 threshold

ters, dozens of increasingly smaller medium-sized ones and a long tail of clusters that contain only two templates.

The vast majority of paraphrase pairs come from sentences that were not originally paraphrases (i.e, sentences that originally had different entities). With a 9.5 threshold, 86% of paraphrases answer that criteria. While that number varies somewhat across thresholds, it is always above 80% and does not consistently increase or decrease as the threshold increases.

	Corpus type	Prec.	PPS
This paper, $\tau = 8$	Unaligned	94%	0.005
This paper, $\tau = 9.5$	Unaligned	82%	0.013
This paper, $\tau = 11$	Unaligned	65%	0.1
Barzilay and McKeown (2001)	Parallel	86.5%	0.1 *
Ibrahim et al. (2003)	Parallel	41.2%	0.11 *
Pang et al. (2003)	Parallel	81.5%	0.33
Barzilay and Lee (2003)	Comparable	78.5%	0.07
Bannard and Callison-Burch (2005)	Parallel bilingual	61.9%	n/a **
Zhao et al. (2009)	Parallel or Comparable	70.6%	n/a **
Wang and Callison-Burch (2011)	Comparable	67%	0.01
Fujita et al. (2012)	Parallel bilingual + unaligned	58%	0.34
Regneri and Wang (2012)	Parallel	79%	0.17

* These papers do not report the number of sentences in the corpus, but do report enough for us to estimate it (e.g. the number of documents or the size in MB)

** These papers do not report the number of paraphrases extracted, or such a number does not exist in their approach

Table 4: Comparison with the precision and paraphrases generated per input sentence (PPS) of relevant prior work

While we wanted to show a meaningful comparison with another method from previous work, none of them do what we are doing here - extraction of sentence-size paraphrasal templates from a non-aligned corpus - and so a comparison using the same data would not be fair (and in most cases, not possible). While it seems that providing the results of human evaluation without comparison to prior methods is the norm in most relevant prior work (Ibrahim et al., 2003; Paşca and Dienes, 2005; Bannard and Callison-Burch, 2005; Fujita et al., 2012), we wanted to at least get some sense of where we stand in comparison to other methods, and so we provide a list of (not directly comparable) results reported by other authors in Table 4.[4] While it is impossible to meaningfully compare and rate such different methods, these numbers support the conclusion that our single-corpus, domain-agnostic approach achieves a precision that is similar to or better than other methods. We also include the *paraphrase per sentence (PPS)* value - the ratio of paraphrases extracted to the number of input sentences of the corpus - for each method in the table. We intend this figure as the closest thing to recall that we can conceive for mining paraphrases. However, keep in mind that it is not a comparable figure across the methods, since different corpora are used. In particular, it is expected to be significantly higher for parallel corpora, where the entire corpus consists of potential paraphrases (and that fact is reflected in Table 4, where some methods that use parallel corpora have a PPS that is an order of magnitude higher than other methods).

8 Discussion and Examples

The first thing to note about the results shown in Figure 1 is that even for the highest threshold considered, which gives us a $\times 21$ improvement in size over the smallest threshold considered, all domains except CWB achieve an average score higher than 3, meaning most of the pairs extracted are paraphrases (CWB is close - a little over 2.9 on average). For the lowest threshold considered, all domains are at a precision above 88%, and for three of them it is 100%. In general, across all domains, there seems to be a significant drop in precision (and a significant boost in size) for thresholds between 9 and 10, while the precisions and sizes are fairly stable for thresholds between 8 and 9 and between 10 and 11. This result is encouraging: since the method seems to behave fairly similarly for different domains with regard to changes in the threshold, we should be able to expect similar behavior for new domains as the threshold is

[4] We always show the results of the best system described. Where needed, if results were reported in a different way than simple percentages, we use averages and other appropriate measures. Some previous work defines related sentences (as opposed to paraphrases) as positives and some does not; we do not change their numbers to fit a single definition, but we use the harsher measure for our own results

adjusted.

The magnitude of precision across domains is another matter. It is clear from the results that some domains are more difficult than others. The Metal domain seems to be the hardest: it never achieves an average score higher than 3.8. For the highest threshold, however, Metal is not different from most of the others, while CWB is significantly lower in precision. The reason seems to be the styles of the domain articles: some domains tend to have a more structured form. For example, each article in the States domain will discuss the economy, demographics, formation etc. of the state, and we are more likely to find paraphrases there (simply by virtue of there being 50×49 candidates). Articles in the Metal domain are much less structured, and there are fewer obvious paraphrase candidates. In CWB articles, there are a few repetitive themes: the outcome of the battle, the casualties, the generals involved etc., but beyond that it is fairly unstructured. This "structurality" of the domain also affects the number of paraphrases that can be found, as evident from the number of paraphrases found in the states domain in Table 3 as compared with the (much larger) Metal and CWB domains.

Table 5 shows a number of examples from each domain, along with the score given to each by the annotators. In an informal error analysis, we saw a few scenarios recurring in low-scored pairs. The Metal example at the bottom of Table 5 is a double case of bad sense disambiguation: the *album* in the second sentence ("Pyromania" in the original) happened to have a name that is also a pathological state. In addition, the *number* in the second sentence really was a date ("1980"). If we had correctly assigned the senses, these two templates would not be paraphrase candidates. The process of grouping by type is an important part of improving precision: two sentences can be misleadingly similar in the vector space, but it is less likely to have two sentences with the exact same entity types and a high vector similarity that are not close in meaning.

Another scenario is the one seen in the NBA example that was scored as 1. Here the senses were chosen correctly, but the level of the hierarchy chosen for the *person* slot was too high. If instead we had chosen *basketball coach* and *basketball player* for the two sentences respectively, they would not be considered as paraphrase candidates (and note that both meanings are implied by the templates). This sort of error does not create a problem (in our evaluation, at least) if the more accurate sense is the same in both sentences - for example, in the other NBA example (which scored 4), the *place* slot could be more accurately replaced with *sports arena* in both templates.

Cases where the types are chosen correctly do not always result in perfect paraphrases, but are typically at least related (e.g. in the examples that scored 2, and to a lesser extent those that scored 3). That scenario can be controlled using a lower threshold, with the downside that the number of paraphrases found decreases.

9 Conclusion and Future Work

We presented a method for extracting paraphrasal templates from a plain text corpus in three steps: templatizing the sentences of the corpus; finding the most appropriate type for each slot; and clustering groups of templates that share the same set of types into paraphrasal sub-groups. We conducted a crowd-sourced human evaluation and showed that our method performs similarly to or better than prior work on mining paraphrases, with three major improvements. First, we do not rely on a parallel or comparable corpus, which are not as easily obtained; second, we produce typed templates that utilize a rich, fine-grained type system, which can make them more suitable for generation; and third, by using such a type system we are able to find paraphrases from sentence pairs that are not, before templatization, really paraphrases.

Many, if not most, of the worst misidentifications seem to be the result of errors in the second stage of the approach - disambiguating the sense and specificity of the slot types. In this paper we focused on a traditional distributional approach that has the advantage of being explainable, but it would be interesting and useful to explore other options such as word embeddings, matrix factorization and semantic similarity metrics. We leave these to future work.

Another task for future work is semantic alignment. Our approach discovers paraphrasal templates without aligning them to a semantic meaning representation; while these are perfectly usable by summarization, question answering, and other text-to-text generation applications, it would be useful for concept-to-text generation and other applications to have each cluster of templates aligned

Score	Domain	Templates
5	States	Per dollar of federal tax collected in [date 1], [american state 1] citizens received approximately [money 1] in the way of federal spending.
		In [date) 1] the federal government spent [money 1] on [american state 1] for every dollar of tax revenue collected from the state.
	AuMa	Designed as a competitor to the [car 1], [car 2] and [car 3].
		It is expected to rival the [car 1], [car 2], and [car 3].
4	CWB	Federal casualties were heavy with at least [number 1] killed or mortally wounded, [number 2] wounded , and [number 3] made prisoner.
		Federal losses were [number 1] killed, [number 2] wounded, and [number 3] unaccounted for – primarily prisoners.
	NBA	For the [date 1] season, the [basketball team 1] moved into their new arena , the [place 1], with a seating capacity of [number 1].
		As a result of their success on the court, the [basketball team 1] moved into the [place 1] in [date 1], which seats over [number 1] fans.
3	Marvel	[imaginary being 1] approached [imaginary being 2], hunting for leads about the whereabouts of the X-Men.
		[imaginary being 1] and [imaginary being 2] eventually found the X-Men and became full time members.
	Metal	In [date 1], [band) 1] recorded their third studio album, "[album 1]", which was produced by Kornelije Kovač.
		[band 1] released their next full-length studio album, "[album 1]" in [date 1].
2	Auma	[company 1] and its subsidiaries created a variety of initiatives in the social sphere, initially in [country 1] and then internationally as the company expanded.
		[company 1] participated in [country 1]'s unprecedented economic growth of the 1950s and 1960s.
	Marvel	Using her powers of psychological deduction, she picked up on [first name 1]'s attraction towards her, and then [first name 2] admits she is attracted to him as well.
		While [first name 1] became shy, reserved and bookish, [first name 2] became athletically inclined, aggressive, and arrogant.
1	NBA	Though the [date 1] 76ers exceeded many on-court expectations, there was a great deal of behind-the-scenes tension between [person 1], his players, and the front office.
		After an [date 1] start, with [person 1] already hurt, these critics seemed to have been proven right.
	Metal	Within [number 1] hours of the statement, he died of bronchial pneumonia, which was brought on as a complication of [pathological state 1].
		With the album's massive success, "[pathological state 1]" was the catalyst for the [number 1] pop-metal movement.

Table 5: Examples of template pairs and their scores

to a semantic representation of the meaning expressed. Since we already discover all the entity types involved, all that is missing is the proposition (or frame, or set of propositions); this seems to be a straightforward, though not necessarily easy, task to tackle in the near future.

References

Gabor Angeli, Percy Liang, and Dan Klein. 2010. A simple domain-independent probabilistic approach to generation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 502–512, Stroudsburg, PA, USA. Association for Computational Linguistics.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: a nucleus for a web of open data. In *Proceedings of the 6th international The semantic web and 2nd Asian conference on Asian semantic web conference*, ISWC'07/ASWC'07, pages 722–735, Berlin, Heidelberg. Springer-Verlag.

Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 597–604, Stroudsburg, PA, USA. Association for Computational Linguistics.

Marco Baroni and Alessandro Lenci. 2010. Distributional memory: A general framework for corpus-based semantics. *Comput. Linguist.*, 36(4):673–721, December.

Regina Barzilay and Lillian Lee. 2003. Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 16–23, Stroudsburg, PA, USA. Association for Computational Linguistics.

Regina Barzilay and Kathleen R. McKeown. 2001. Extracting paraphrases from a parallel corpus. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, ACL '01, pages 50–57, Stroudsburg, PA, USA. Association for Computational Linguistics.

Rahul Bhagat and Deepak Ravichandran. 2008. Large Scale Acquisition of Paraphrases for Learning Surface Patterns. In *Proceedings of ACL-08: HLT*, pages 674–682, Columbus, Ohio, June. Association for Computational Linguistics.

William Blacoe and Mirella Lapata. 2012. A comparison of vector-based representations for semantic composition. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 546–556, Stroudsburg, PA, USA. Association for Computational Linguistics.

Nathanael Chambers and Dan Jurafsky. 2011. Template-based information extraction without the templates. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 976–986, Stroudsburg, PA, USA. Association for Computational Linguistics.

Dipanjan Das and Noah A. Smith. 2009. Paraphrase identification as probabilistic quasi-synchronous recognition. In *Proc. of ACL-IJCNLP*.

Bill Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of Coling 2004*, pages 350–356, Geneva, Switzerland, Aug 23–Aug 27. COLING.

Daniel Duma and Ewan Klein. 2013. Generating natural language from linked data: Unsupervised template extraction. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013) – Long Papers*, pages 83–94. ASSOC COMPUTATIONAL LINGUISTICS-ACL.

Christiane Fellbaum, editor. 1998. *WordNet An Electronic Lexical Database*. The MIT Press.

Atsushi Fujita, Pierre Isabelle, and Roland Kuhn. 2012. Enlarging paraphrase collections through generalization and instantiation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 631–642, Stroudsburg, PA, USA. Association for Computational Linguistics.

Weiwei Guo and Mona Diab. 2012. Modeling sentences in the latent space. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, ACL '12, pages 864–872, Stroudsburg, PA, USA. Association for Computational Linguistics.

Marti A Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics-Volume 2*, pages 539–545. Association for Computational Linguistics.

Ali Ibrahim, Boris Katz, and Jimmy Lin. 2003. Extracting structural paraphrases from aligned monolingual corpora. In *Proceedings of the Second International Workshop on Paraphrasing - Volume 16*, PARAPHRASE '03, pages 57–64, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yangfeng Ji and Jacob Eisenstein. 2013. Discriminative improvements to distributional sentence similarity. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 891–896. ACL.

Ravi Kondadadi, Blake Howald, and Frank Schilder. 2013. A statistical nlg framework for aggregated planning and realization. In *ACL (1)*, pages 1406–1415. The Association for Computer Linguistics.

Dekang Lin and Patrick Pantel. 2001. Dirt @sbt@discovery of inference rules from text. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '01, pages 323–328, New York, NY, USA. ACM.

N. Madnani, Philip Resnik, Bonnie J Dorr, and R. Schwartz. 2008. Applying automatically generated semantic knowledge: A case study in machine translation. *NSF Symposium on Semantic Knowledge Discovery, Organization and Use*.

Nitin Madnani, Joel Tetreault, and Martin Chodorow. 2012. Re-examining machine translation metrics for paraphrase identification. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL HLT '12, pages 182–190, Stroudsburg, PA, USA. Association for Computational Linguistics.

Rada Mihalcea, Courtney Corley, and Carlo Strapparava. 2006. Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1*, AAAI'06, pages 775–780. AAAI Press.

D. Milajevs, D. Kartsaklis, M. Sadrzadeh, and M. Purver. 2014. Evaluating neural word representations in tensor-based compositional settings. In *Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar. Association for Computational Linguistics, Association for Computational Linguistics.

Marius Paşca and Péter Dienes. 2005. Aligning needles in a haystack: Paraphrase acquisition across the web. In *Proceedings of the Second International Joint Conference on Natural Language Processing*, IJCNLP'05, pages 119–130, Berlin, Heidelberg. Springer-Verlag.

Bo Pang, Kevin Knight, and Daniel Marcu. 2003. Syntax-based alignment of multiple translations: Extracting paraphrases and generating new sentences. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 102–109, Stroudsburg, PA, USA. Association for Computational Linguistics.

Chris Quirk, Chris Brockett, and William Dolan. 2004. Monolingual machine translation for paraphrase generation. In *In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 142–149.

Michaela Regneri and Rui Wang. 2012. Using discourse information for paraphrase extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 916–927, Stroudsburg, PA, USA. Association for Computational Linguistics.

Satoshi Sekine. 2005. Automatic paraphrase discovery based on context and keywords between ne pairs. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, pages 80–87.

Satoshi Sekine. 2006. On-demand information extraction. In *Proceedings of the COLING/ACL on Main Conference Poster Sessions*, COLING-ACL '06, pages 731–738, Stroudsburg, PA, USA. Association for Computational Linguistics.

Siwei Shen, Dragomir R. Radev, Agam Patel, and Güneş Erkan. 2006. Adding syntax to dynamic programming for aligning comparable texts for the generation of paraphrases. In *Proceedings of the COLING/ACL on Main Conference Poster Sessions*, COLING-ACL '06, pages 747–754, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yusuke Shinyama, Satoshi Sekine, and Kiyoshi Sudo. 2002. Automatic paraphrase acquisition from news articles. In *Proceedings of the Second International Conference on Human Language Technology Research*, HLT '02, pages 313–318, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, and Christopher D. Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 151–161, Stroudsburg, PA, USA. Association for Computational Linguistics.

Rui Wang and Chris Callison-Burch. 2011. Paraphrase fragment extraction from monolingual comparable corpora. In *Proceedings of the 4th Workshop on Building and Using Comparable Corpora: Comparable Corpora and the Web*, BUCC '11, pages 52–60, Stroudsburg, PA, USA. Association for Computational Linguistics.

Shiqi Zhao, Xiang Lan, Ting Liu, and Sheng Li. 2009. Application-driven statistical paraphrase generation. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*, ACL '09, pages 834–842, Stroudsburg, PA, USA. Association for Computational Linguistics.