Essentia: Mining Domain-specific Paraphrases with Word-Alignment Graphs

Danni Ma[1], Chen Chen[2], Behzad Golshan[2], Wang-Chiew Tan[2]
[1]Department of Computer and Information Science, University of Pennsylvania
[2]Megagon Labs
dannima@seas.upenn.edu, {chen,behzad,wangchiew}@megagon.ai

Abstract

Paraphrases are important linguistic resources for a wide variety of NLP applications. Many techniques for automatic paraphrase mining from general corpora have been proposed. While these techniques are successful at discovering generic paraphrases, they often fail to identify domain-specific paraphrases (e.g., {"*staff*", "*concierge*"} in the hospitality domain). This is because current techniques are often based on statistical methods, while domain-specific corpora are too small to fit statistical methods. In this paper, we present an unsupervised graph-based technique to mine paraphrases from a small set of sentences that roughly share the same topic or intent. Our system, ESSENTIA, relies on word-alignment techniques to create a *word-alignment graph* that merges and organizes tokens from input sentences. The resulting graph is then used to generate candidate paraphrases. We demonstrate that our system obtains high quality paraphrases, as evaluated by crowd workers. We further show that the majority of the identified paraphrases are domain-specific and thus complement existing paraphrase databases.

1 Introduction

Paraphrases are important linguistic resources which are widely used in many NLP tasks, including text-to-text generation (Ganitkevitch et al., 2011), recognizing textual entailment (Dagan et al., 2005), and machine translation (Marton et al., 2009). Today, mining paraphrases still remains an active research area (Ferreira et al., 2018; Gupta et al., 2018; Iyyer et al., 2018; Zhang et al., 2019). Most existing work on this topic focuses on mining general-purpose paraphrases (e.g., {"*prevalent*", "*very common*"}), but fails to extract **domain-specific paraphrases**. For example, while {"*reservation*", "*stay*"} are not paraphrases in general, they are interchangeable in the following sentence:

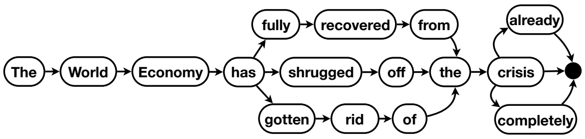

Figure 1: An instance of a word-alignment graph.

Can we extend our reservation for two more days?

Existing paraphrase mining techniques are often based on statistical methods. They cannot be immediately applied to domain-specific corpora, because such corpora are usually smaller in size and lack parallel data. ESSENTIA overcomes this problem by using an unsupervised graph-based method that mines domain-specific paraphrases from a small set of short sentences sharing the same topic or intent. ESSENTIA's key insight is that a collection of sentences from a specific domain often exhibit common patterns. ESSENTIA makes use of these properties to align tokens of input sentences. The resulting alignments are then summarized in a directed acyclic graph (DAG) called the *word-alignment graph*. It illustrates which phrases can be used interchangeably and thus are potential paraphrases. Figure 1 shows the word-alignment graph generated from the following three sentences:

- *The world economy has fully recovered from the crisis.*
- *The world economy has shrugged off the crisis completely.*
- *The world economy has gotten rid of the crisis already.*

The word-alignment graph reveals that phrases that are not aligned, but share the same aligned context (i.e. surrounding words) are likely to be domain-specific paraphrases. Hence, even though {"*fully recovered from*", "*shrugged off*", "*gotten rid of*"} are not aligned, they are likely paraphrases because they share the same patterns be-

Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13), pages 52–57
Hong Kong, November 4, 2019. ©2019 Association for Computational Linguistics

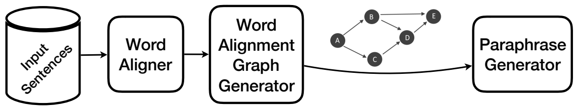

Figure 2: The architecture of ESSENTIA.

fore and after themselves.

While this work is focused on mining paraphrases, we believe that word-alignment graphs have other interesting applications, and we leave them for future work. For instance, a word-alignment graph enables one to generate new sentences or phrases that do not appear in the original set of sentences. "*The world economy has gotten rid of the crisis completely*" is a new sentence that is generated using the graph in Figure 1.

Contributions. We present ESSENTIA, an unsupervised system for mining domain-specific paraphrases by creating rich graph structures from small corpora. Experiments on datasets in real-world applications demonstrate that ESSENTIA finds high-quality domain-specific paraphrases. We also validate that these domain-specific paraphrases complement and augment PPDB (Paraphrase Database), the most extensive paraphrase database available in the community.

2 Essentia

The architecture of ESSENTIA (Figure 2) consists of: (1) a word aligner which aligns similar words (and phrases) between different sentences based on syntactic and semantic similarity; (2) a word-alignment graph generator that summarizes the alignments into a compact graph structure; and (3) a paraphrase generator that mines domain-specific paraphrases from the word-alignment graph. We describe each component below.

2.1 Word aligner

We use the state-of-the-art monolingual word aligner by Sultan et al. (2014). The input to the word aligner is a single pair of sentences and the output is a predicted mapping between tokens of two sentences. ESSENTIA uses the word aligner to compute the alignments for all pairs of sentences provided as input.

Every sentence is first pre-processed by replacing numbers and named entities – which are identified by spaCy (Honnibal and Montani, 2017) – with special symbols "NUM" and "ORG" respectively before it is passed to the word aligner.

The word aligner relies on paraphrase, lexical resources and word embedding techniques to find a mapping between tokens. In other words, the word aligner finds general-purpose paraphrases and maps their tokens accordingly. ESSENTIA further processes the output of the word aligner to mine domain-specific paraphrases.

2.2 Word-alignment graph generator

Once the alignments between every pair of sentences are available, the word-alignment graph generator summarizes all the alignments into a unified structure, referred to as the word-alignment graph. It is a DAG that represents all the input sentences (see Figure 1 as an example). The process of creating the word-alignment graph is described as follows.

The first step partitions the set of input sentences into *compatible* groups. A group of sentences is compatible if their alignments adhere to the following three conditions:

- **Injectivity** For any pair of sentences, each word should be mapped to at most one word in the other sentence.
- **Monotonicity** For any pair of sentences, if a word $w1$ appears before $w2$, then the word that $w1$ maps to should also appears before the word that $w2$ maps to in the other sentence. Sentence pairs such as "*Yesterday I saw him*" and "*I saw him yesterday*" violate this condition.
- **Transitivity** Given any three sentences s_1, s_2, and s_3, if a word w_1 in s_1 is mapped to w_2 in s_2, and w_2 in s_2 is mapped to w_3 in s_3, then w_1 should be only mapped to w_3 in s_3.

The above conditions are necessary to ensure that the resulting representation is compact and forms a DAG. We start by partitioning the input sentences into compatible groups. The partitioning strategy is a simple greedy algorithm which starts with a single empty group. A sentence will be added to the first group that remains compatible upon adding this new sentence. If no such group exists, a new empty group is created and the sentence is added to this group. This process repeats until each sentence is assigned to one group.

Next, the word-alignment graph generator represents each group as a DAG and then combines all the DAGs using a shared start-node and end-node to create the final word-alignment graph. Specifically, a line graph is first created for each sentence (i.e., a word-alignment graph for a single sentence). Then, the alignments are processed: for each pair of aligned words, their corresponding nodes are contracted to a single node. Due to the constraints imposed earlier, one can easily show that the resulting graph will be cycle-free.

2.3 Paraphrase generator

Given a word-alignment graph, the paraphrase generator considers all paths in the graph that share the same start and end node as paraphrase candidates. For instance, in Figure 1, there are three branches that start from the node "*has*" and end in "*the*". Consequently, the phrases {"*fully recovered from*", "*shrugged off*", "*gotten rid of*"} are extracted as paraphrase candidates.

However, not all extracted candidates are paraphrases. Consider the following sentences:

- Give me directions to my parent's place

- Give me directions to the Time Square

In this case, {"*my parent's place*", "*the Time Square*"} will be extracted as candidates, but it is clear that they are not valid paraphrases.

To avoid generating wrong paraphrases, we design a filtering step – which can be implemented either using rules (e.g., regular expressions) or statistical methods (e.g., word similarity) – on top of the extracted candidates. Our current implementation of this filtering functionality adopts a rule-based heuristic that only considers candidates of verb phrases containing three or fewer tokens, such as {"*access to Wi-Fi*", "*hookup to Wi-Fi*"}. Our empirical study reveals that many such verb phrases are domain-specific paraphrases. Other classes of phrases, such as noun phrases, turn out to have much noise. For example, many noun phrases are simply different options (e.g., {"*today*","*tomorrow*"}). We leave the design of advanced filters for those classes as future work.

In the process of discovering paraphrases, we observe that sentences can be "cleaned". That is, some phrases can be removed without affecting the essential meaning of a sentence. Figure 1 shows that the phrases "*already*" and "*completely*" share the same start and end node. Moreover, we see that the start and end node are also directly connected with a single edge. Such phrases are *optional phrases* and can be removed without affecting the core meaning of a sentence. By identifying optional phrases, we can simplify the set of input sentences to its "essence", where the name of ESSENTIA comes from.

Notes on scalability. The time required by the word aligner to compute alignments between two sentences is quite small and can be considered as constant since the length of input sentences is bounded in practice. Given that, the time-complexity of ESSENTIA's pipeline for n input sentences is $O(n^2)$ as we need to compute alignments between all pairs of sentences. In practice, the pipeline can be applied to roughly a hundred sentences within an hour. For a larger collection of sentences, as described in Section 2.2, we first run a clustering algorithm to group sentences into smaller clusters, and then feed each cluster to ESSENTIA's pipeline.

3 Related Work

Collecting and curating a database of paraphrases is a costly and time-consuming task in general. Although there are existing techniques to collect paraphrase pairs from crowd-workers more efficiently and with lower cost (Chen and Dolan, 2011), there has been a great interest in developing techniques for automatically mining paraphrases from existing corpora. Barzilay and McKeown (2001) proposed the first unsupervised learning algorithm for paraphrase acquisition from a corpus of multiple English translations of the same source text. Barzilay and Lee (2003) followed up with an approach that applied multiple-sequence alignment to sentences gathered from parallel corpora. Pang et al. (2003) proposed a new syntax-based algorithm to produce word-alignment graphs for sentences. Finally, Quirk et al. (2004) applied statistical machine translation techniques to extract paraphrases from monolingual parallel corpora.

The most extensive resource for paraphrases today is PPDB (Ganitkevitch et al., 2013; Ganitkevitch and Callison-Burch, 2014; Pavlick et al., 2015b). PPDB consists of a huge number of phrase pairs with confidence estimates, and has already been proven effective for multiple tasks. However, as our experiments show, PPDB and other resources fail to capture a large number of domain-specific paraphrases.

To extract domain-specific paraphrases, Pavlick et al. (2015a) extended Moore-Lewis method (Moore and Lewis, 2010) and learned paraphrases from bilingual corpora. Zhang et al. (2016) constructed Markov networks of words and picked paraphrases based on the frequency

	Dataset	# of extracted pairs	# of valid pairs	Precision
ESSENTIA	Snips	173	84	48.55%
	HotelQA	2221	642	28.91%
FSA	Snips	18	15	83.33%
	HotelQA	342	185	54.09 %

Table 1: Comparison between ESSENTIA and FSA baseline on paraphrase extraction

of co-occurrences. However, these systems rely on significantly large amounts of domain-specific data (either for supervised training or conducting frequency analysis), which may not always be available. ESSENTIA instead uses an unsupervised graph-based technique for paraphrase mining and does not rely on the presence of a large amount of domain-specific data. The word-alignment graph constructed by ESSENTIA can be interpreted as an extension of multi-sentence compression (Filippova, 2010). We compactly maintain all paths and expressions in the constructed word-alignment graphs. As pointed out in Pang et al. (2003), the extracted paraphrases can help enrich the diversity of expressions regarding a specific intention, and ultimately provide more training examples for data-driven models.

4 Evaluation

ESSENTIA is evaluated on two datasets and is shown to generate high quality domain-specific paraphrases. We compare our system against a syntax-based alignment technique by Pang et al. (2003), which we refer to as FSA, as it generates *Finite-State Automata* for compactly representing sentences in a setting similar to ours. Compared to FSA, ESSENTIA generates 263% more paraphrases on those two datasets. We further demonstrate that most extracted paraphrases are truly domain-specific and thus are missing from PPDB.

Datasets We use two datasets to evaluate ESSENTIA. The first one, commonly known as the Snips dataset (Coucke et al., 2018), is a collection of queries submitted to smart conversational devices (e.g., Google Home or Alexa). Snips has ten documents, each covering one intent such as "*Get Directions*", "*Get Weather*" and so on. On average, each document has 32 sentences, and each sentence has 9 words. The other dataset – which is called HotelQA – is an industry proprietary dataset of various types of questions submitted by hotel guests regarding different amenities and services, such as "*Check-out*" or "*Wi-Fi*". HotelQA also consists of ten documents, with an average of 54 sentences per document and 10 words per sentence. HotelQA was our primary motivation for investigating this problem. The industry application requires an automatic method to identify a set of questions that are semantically equivalent.

4.1 Mining Paraphrases

Table 1 compares the performance of ESSENTIA with the FSA baseline for paraphrase mining. Specifically, we show the number of phrase pairs extracted by ESSENTIA and FSA from both datasets ("# of extracted pairs" column), number of valid paraphrases within these pairs ("# of valid pairs" column), and precision ("Precision" column). Although FSA has higher precision due to conservative sentence alignment, ESSENTIA extracts significantly more paraphrases, improving the recall by 460% (Snips) and 247% (HotelQA) over the baseline. To identify valid paraphrases, we design a crowd-sourcing task on Figure-Eight Data Annotation Platform. In this task, we present an extracted candidate pair (e.g., {*"log onto"*, *"connect to"*}) and a domain (e.g., "Wi-Fi") to human annotators, and ask them to decide whether the two phrases are paraphrases or not.

ESSENTIA discovers a large number of paraphrases missing from PPDB, which has the highest coverage among the existing paraphrase resources (Pavlick and Callison-Burch, 2016). More precisely, we take the 726 correct extractions of ESSENTIA (as verified by human annotators) and search to see if they appear in PPDB even with low confidence scores. We find that only **4%** of our discovered paraphrases appear in PPDB. This in turn shows the effectiveness of ESSENTIA in discovering paraphrases, because it goes beyond PPDB by using only a few sentences. Table 2 lists some domains and examples of domain-specific paraphrases detected by ESSENTIA.

Finally, to better understand how ESSENTIA's performance can be improved and what opportunities lie ahead for further research, we review a sample of ESSENTIA's incorrect extractions and identify two major classes of errors. One class

Domain	Example paraphrases
Restaurant search	recommend a good place suggest a place
Restaurant reservation	get me a place get me a spot
Get directions	show me the way get me directions
Get weather	need the weather want the weather
Request ride	find a taxi need an uber
Share location	share my location send my location
Hotel Wi-Fi	log onto the Wi-Fi connect to Wi-Fi
Hotel checkout	extend our checkout have a late checkout

Table 2: Examples of domain-specific paraphrases.

consists of expressions that are alternative options but not necessarily paraphrases (e.g., {*"avoiding the highway"*, *"avoiding toll road"*}). Another class contains expressions that involve the same topic but have slightly different intentions (e.g., {"*tell me the Wi-Fi password*", "*how to connect to Wi-Fi*"}). While the two error classes we discuss here are the most prevalent ones, an in-depth analysis of error classes and their frequencies (which we leave as future work) can be quite insightful.

5 Conclusion and Future Work

We present ESSENTIA, an unsupervised graph-based system for extracting domain-specific paraphrases, and demonstrate its effectiveness using datasets in real-world applications. Empirical results show that ESSENTIA can generate high quality domain-specific paraphrases that are largely absent from mainstream paraphrase databases.

Future work involves various directions. One direction is to derive domain-specific sentence templates from corpora. These templates can be useful for natural language generation in question-answering systems or dialogue systems. Second, the current method can be extended to mine paraphrases from a wide range of syntactic units other than verb phrases. Also, the word aligner can be improved to align prepositions more accurately, so that the generated alignment graph would reveal more paraphrases. Finally, ESSENTIA can also be used to identify linguistic patterns other than paraphrases, such as phatic expressions (e.g., "*Excuse me*", "*All right*"), which will in turn allow us to identify the essential constituents of a sentence.

References

Regina Barzilay and Lillian Lee. 2003. Learning to paraphrase: an unsupervised approach using multiple-sequence alignment. In *Proceedings of NAACL-HLT 2003*.

Regina Barzilay and Kathleen R McKeown. 2001. Extracting paraphrases from a parallel corpus. In *Proceedings of NAACL-HLT 2003*, pages 50–57.

David L. Chen and William B. Dolan. 2011. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of ACL 2011*, pages 190–200.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The PASCAL recognising textual entailment challenge. In *Machine Learning Challenges Workshop*, pages 177–190.

Rafael Ferreira, George DC Cavalcanti, Fred Freitas, Rafael Dueire Lins, Steven J Simske, and Marcelo Riss. 2018. Combining sentence similarities measures to identify paraphrases. *Computer Speech & Language*, pages 59–73.

Katja Filippova. 2010. Multi-sentence compression: Finding shortest paths in word graphs. In *Proceedings of COLING 2010*, pages 322–330.

Juri Ganitkevitch and Chris Callison-Burch. 2014. The multilingual paraphrase database. In *Proceedings of LREC 2014*, pages 4276–4283.

Juri Ganitkevitch, Chris Callison-Burch, Courtney Napoles, and Benjamin Van Durme. 2011. Learning sentential paraphrases from bilingual parallel corpora for text-to-text generation. In *Proceedings of EMNLP 2011*, pages 1168–1179.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The paraphrase database. In *Proceedings of NAACL-HLT 2013*, pages 758–764.

Ankush Gupta, Arvind Agarwal, Prawaan Singh, and Piyush Rai. 2018. A deep generative framework for paraphrase generation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of NAACL-HLT 2018*, pages 1875–1885.

Yuval Marton, Chris Callison-Burch, and Philip Resnik. 2009. Improved statistical machine translation using monolingually-derived paraphrases. In *Proceedings of EMNLP 2009*, pages 381–390.

Robert C Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of ACL 2010*, pages 220–224. Association for Computational Linguistics.

Bo Pang, Kevin Knight, and Daniel Marcu. 2003. Syntax-based alignment of multiple translations: Extracting paraphrases and generating new sentences. In *Proceedings of NAACL-HLT 2003*, pages 102–109.

Ellie Pavlick and Chris Callison-Burch. 2016. Simple PPDB: A paraphrase database for simplification. In *Proceedings of ACL 2016*, pages 143–148.

Ellie Pavlick, Juri Ganitkevitch, Tsz Ping Chan, Xuchen Yao, Benjamin Van Durme, and Chris Callison-Burch. 2015a. Domain-specific paraphrase extraction. In *Proceedings of ACL-IJCNLP 2015*, pages 57–62.

Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2015b. Ppdb 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *Proceedings of ACL-IJCNLP 2015*, pages 425–430.

Chris Quirk, Chris Brockett, and William Dolan. 2004. Monolingual machine translation for paraphrase generation. In *Proceedings of EMNLP 2004*, pages 142–149.

Md. Arafat Sultan, Steven Bethard, and Tamara Sumner. 2014. Back to basics for monolingual alignment: Exploiting word similarity and contextual evidence. *Transactions of the Association for Computational Linguistics*, pages 219–230.

Lilin Zhang, Zhen Weng, Wenyan Xiao, Jianyi Wan, Zhiming Chen, Yiming Tan, Maoxi Li, and Mingwen Wang. 2016. Extract domain-specific paraphrase from monolingual corpus for automatic evaluation of machine translation. In *Proceedings of WMT 2016*, pages 511–517.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. In *Proceedings of NAACL-HLT 2019*, pages 1298–1308.