Domain Specific Term Extraction on Reddit corpus

Matthew Luettgen

12/17/2020

Abstract

In this project, I aim to answer if the social media platform Reddit is a valid corpus for domain-specific term extraction. Using chunking of noun phrases and TF-IDF, terms are extracted from a small yet varying selection of subreddit corpora. This project is opens the door for detecting domain-specific words of an enormous amount of domains outside of informative documents.

1 Introduction

1.1 Domain-Specific Terms

This project aims to create a prototype extract domain-specific terms from social media text corpora. Kim and Cavedon (2011) define domain-specific terms as "terms that have significant meaning(s) in a specific domain." For example, the word "word2vec" is a term specific to the domain of Natural Language Processing. Also, domain specific terms take the form of words that only differ in meaning. For example "RAM" in the domain of computer science means "Random Access Memory" and not the animal "Ram." I define a domain as text written about topic by a community of people knowledgeable about that topic. Often, domain-specific terms are terms that would have to be explained to someone outside the community.

1.2 Previous Work

Park et al. (2008) first extracted key words of documents with the intention of categorizing documents to domains with a support vector machine (SVM). They created a simple formula to extract keywords: the count of a term in the domain-specific corpus over the length of the domain specific-corpus divided by the count of the term in the entire corpus over the length of the entire corpus. Although formula is similar to Term-Frequency Inverse-Document-Frequency (TF-IDF), Kim et al. (2009) expanded upon the method and attained better results by using TF-IDF. Both these studies used these measures to weight terms in an SVM that categorized text into documents. Other approaches use the structure of websites to determine domain specific terminology. Vivaldi and Rodríguez (2010) used Wikipedia to extract domain-specific terminology, using a metric related to the layout of Wikipedia to define domains and domain-specific terms. Kim and Cavedon (2011) used Wiktionary with n-grams, domain concepts, and topics of terms to train an SVM to classify terms. Meyers et al. (2014) used part-of-speech tagging and a finite state machine to divide text into chunks, then used rule-based boolean measures to determine domain-specific terms.

1.3 Significance of Domain-Specific Term Extraction

Domain-specific term extraction is often helpful in the categorization of documents such as in Kim et al. (2009) or Park et al. (2008). Liu and Cai (2015) found that domain specific terms can also provide a challenge to machine translation and that detecting them can assist machine translation algorithms. More specific to the advantages of using a social media corpus, using domain-specific terms of a particular group makes one seem like an in-member of a group. Therefore, this technology may potentially be useful for advertisers looking to customize ads to a specific groups. Furthermore, if one were to extract domain-specific terms on forums similar to reddit focusing on criminal activity, these terms could potentially be useful to law-enforcement.

1.4 Social Media Corpus

The majority of previous projects such as Meyers et al. (2014), Kim and Cavedon (2011), Vivaldi and Rodríguez (2010) have used more academic domains, being patents, Wikitionary (an online dictionary), and Wikipedia pages about chemistry and astronomy. However this project intends to extend similar methods to a wider variety of less formal domains by using a social media corpus via Reddit. Reddit allows any user to create a page about a topic called a subreddit where other users may post and comment on that topic. Allowing a subreddit to represent a single domain, Reddit data provides domains of a great amount of topics. This allows domain-specific term extraction of domains that cover geographic communities, hobbies and pastimes, and any other common interest. In contrast to previous corpora in similar projects, Reddit contains posts and comments in a forum rather than one user writing a single document. This paper aims to answer if social media and Reddit in particular is a viable corpus to extract domain-specific terms.

2 Methods

2.1 Reddit Corpus

The ConvoKit developers at Cornell collected a text corpus containing posts and comments from Reddit sorted by subreddit. The entire corpus includes over 900,000 different subreddits with every post and comment from the beginning of the subreddit until the date the data was collected in October 2018. Each subreddit is able to be downloaded individually. For a more manageable corpus, convokit developers also also collected a smaller corpus of 100 comment threads in 100 popular subreddits called reddit-corpus-small. This smaller collection is used as the base corpus to be compared to while larger subreddits downloaded from the entire collection serve as the corpus test extracting domain-specific vocabulary.

2.2 Chunking Noun Phrases

As Meyers et al. (2014) found, many domain-specific terms take the form of long noun phrases such as "Domain-specific term extraction." Therefore, a similar though less complex process of chunking Meyers et al. (2014) was applied to the data. Instead of a finite state machine, dependency parsing using spaCy was used to chunk data. Sentences were parsed using the sentence tokenizer in nltk. Then, for each noun phrase (NP) in the sentence, the representation of the NP's subtree, then the representation of all of the phrases within the subtree to the right and left of the tree were treated as potential terms. For example, the NP "very big red car parked in the garage that goes fast," the entire NP, "very big red car," "red car," "car parked in the garage," and "car parked in the garage that goes fast" would all be collected as potential terms, though the length limit was set to a maximum of six words to save on computation time.

2.3 TF-IDF

Term-Frequency Inverse-Document-Frequency is an algorithm often used in information retrieval to rank documents based on the relevancy of a term in a document. Term Frequency represents the frequency of the term in a document and Inverse Document Frequency represents the amount of documents that contain the term. In this project, terms were considered to be all unigrams extracted with the word tokeinzer in nltk and the noun phrases from the previous step. Similar to Kim et al. (2009) this project uses this algorithm to rank terms based on how relevancy of the term in the document, where one document is a subreddit being compared to other subreddits. There are many different variants of TF-IDF though after testing several different variants the variant depicted below was chosen.

$$\frac{\log\left(1+\ C(t,d)\right)}{Length(d)} * \log\left(1+\frac{N-k(t)}{k(t)}\right)$$

c(t,d): count of term t in document d

Length(d): length of document d

N = number of documents in corpus

k(t) = count of documents that contain term t

Terms were ranked based on TF-IDF scores for evaluation. A threshold metric was considered, however for evaluation I classify a certain number of standard deviations over the mean as domain-specific terms, depending on the subreddit, aiming for a total amount of between two-hundred and two-hundred-fifty terms.

3 Results

Other studies such as Kim et al. (2009) have had evaluators scan over the corpus and manually pick out domain-specific-terms then have other participants mark each term in results as domain-specific or non-domain-specific. However, for this small project I do not have the money nor the time to hire people to evaluate results so therefore I evaluated the results myself. I did not have the time available to scan over the corpus and pick out domain-specific vocabulary which limited the evaluation to only precision as accuracy and recall depend on knowledge of false negatives.

Three subreddits were chosen for evaluation, each meant to vary greatly in content matter. Since I evaluated results myself I chose subreddits of topics in which I personally was knowledgeable. First, r/LanguageTechnology, which mostly contains users asking questions about various NLP subjects, was chosen as an example of a more information-based corpus with many technical terms. In contrast, r/StarWarsEU, which contains users discussing the Star Wars Expanded Universe and contains various sci-fi and fantasy jargon, was chosen as a hobby/interest based subreddit completely removed from academia. Finally, r/IndianaUniversity, which contains mostly questions about IU campus life, was chosen as a domain that covers a narrow group of users joined by a university and also a geographic region instead of a common interest. Each subreddit contains between 30,000 and 111,000 sentences.

Precision	LT	SWEU	IU	IU+college
Precision@5	1.0	1.0	1.0	1.0
Precision@25	0.96	0.96	0.88	1.0
Precision@50	0.96	0.96	0.86	1.0
Precision@100	0.98	0.97	0.77	.92
Precision@150	0.96	0.96	0.77	.93
Precision@200	0.94	0.94	0.74	.9
Precision@all	0.94	0.94	0.73	.91

Table 1: Results
LT = r/LangaugeTechnology
SWEU = r/StarWarsEU
IU = r/IndianaUniversity
IU + college = r/IndianaUniversity + college-specific terms

The algorithm performed very successfully, considering that precision was the only metric of evaluation. For r/LanguageTechnology retrieved the names of algorithms, applications, and people relating to natural language processing. For r/StarWarsEU, the algorithm pulled up book titles, authors, characters, and sci-fi terms. And for r/IndianaUniversity, the algorithm extracted buildings, class numbers, programs, organizations, and restaurants at IU. Overall, it was very successful as each algorithm was able to attain precision of at least .7.

Manually evaluating the results exposed two major problems. The first is mistakes in chunking. Although "Natural Langauge Processing" is a valid domain-specific term in the subreddit r/LanguageTechnology, however "Language processing" really isn't. However, both terms were brought up. To solve this problem, I would further improve the chunking process by possibly statistically comparing the cut-word to the whole word. The second mistake brings up questions about the definition of "domain" and how it should be interpreted. Although the algorithm performed somewhat poorly when defining the domain as only students at IU, many of the terms apply more

broadly to college life in general. For example "Kelley Student," referring to students attending the Kelley School of Business is a valid domain-specific term. However, "Business school" which can apply to any University's business school was also extracted by the algorithm. On one hand, the domain of IU students can be though as a sub-domain of university students. The algorithm performed significantly better when the domain was considered to include general college life terms. On the other hand, this also may be a restriction of a small corpus size, as there was no other university subreddit in the base corpus. A larger corpus would benefit these domain mistakes.

4 Discussion

In this project, we have proven that Reddit is valid source for gathering domain-specific terms of a wide variety of domains ranging from hobbies and interests to geographic locations to technical topics. If one is familiar with the domain being tested, one can observe that the terms extracted fairly accurately represent the domain by simply skimming through the results. The number of domains outside of this project this algorithm can extract terms from is immense. As Reddit contains over 900,000 subreddits, one can find domain-specific terms for any domain as long as the domain is represented by at least one of these subreddits.

To further improve this algorithm, I would first create a larger corpus to prevent the domain mistakes presented earlier. The corpus I used contained only very popular subreddits. Ideally, every single subreddit would be used in the base-corpus though this would not be efficient in processing time or disk space. However, a broad sample from a larger variety of subreddits would certainly benefit the algorithm.

I can think of several directions to continue this project. One continuation is that not only would the algorithm be able to extract the terms, but also to provide information on domain-specific terms so that an outsider may quickly have insight on the meaning of the terms. This information could contain the part of speech, synonyms, related words for each term, or even a possible definition. Another possible application of this is similar to Kim

and Cavedon (2011) who used domain-specific words to classify documents. If one considers the text from a user of any social media to be a document, then one could possibly classify the user by the domains they belong to by detecting domain-specific words extracted by this algorithm in their text. This algorithm and dataset opens many doors for later possibilities.

References

Su Nam Kim and Lawrence Cavedon. Classifying domain-specific terms using a dictionary. In *Proceedings of the Australasian Language Technology Association Workshop 2011*, pages 57–65, Canberra, Australia, December 2011. URL `https://www.aclweb.org/anthology/U11-1009`.

Su Nam Kim, Timothy Baldwin, and Min-Yen Kan. Extracting domain-specific words - a statistical approach. In *Proceedings of the Australasian Language Technology Association Workshop 2009*, pages 94–98, Sydney, Australia, December 2009. URL `https://www.aclweb.org/anthology/U09-1013`.

Weisong Liu and Shu Cai. Translating electronic health record notes from English to Spanish: A preliminary study. In *Proceedings of BioNLP 15*, pages 134–140, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.18653/v1/W15-3816. URL `https://www.aclweb.org/anthology/W15-3816`.

Adam Meyers, Zachary Glass, Angus Grieve-Smith, Yifan He, Shasha Liao, and Ralph Grishman. Jargon-term extraction by chunking. In *Proceedings of the COLING Workshop on Synchronic and Diachronic Approaches to Analyzing Technical Language*, pages 11–20, Dublin, Ireland, August 2014. Association for Computational Linguistics and Dublin City University. doi: 10.3115/v1/W14-6002. URL `https://www.aclweb.org/anthology/W14-6002`.

Youngja Park, Siddharth Patwardhan, Karthik Visweswariah, and Stephen

Gates. An empirical analysis of word error rate and keyword error rate. pages 2070–2073, 01 2008.

Jorge Vivaldi and Horacio Rodríguez. Finding domain terms using wikipedia. 12 2010.