Matt Luettgen

Domain-Specific Term Extraction usingTF-IDF for Reddit corpus.

Introduction and Motivation:

Kim and Cavedon (2009) define domain-specific terms as “terms that have significant meaning(s) in a specific domain.” Although “domain” is often implied to mean a particular academic field such as medicine, here I am defining a “domain” as text coming from any particular community of speakrs. For example, in this paper I am considering “Krogucchi” and “Kroghetto” to be terms specific to the domain of speech coming from people living in Bloomington, Indiana. A domain-specific may also only differ from general English by usage or meaning. For example, in the domain of computer science, “RAM” meaning “Random Access Machine” has a different meaning than the animal “ram” in general English.

Automatic domain-specific term extraction has uses in other tasks such as keyphrase extraction, word sense disambiguation, and query expansion (Kim et al. 2009). However, there is a lack of research on domain-specific term extraction on social media, instead researchers have used technical documents (Meyes et al 2014), medical records (Weisong et al. 20115), or Wikipedia (Vivaldi et al. 2010). They key difference here is that in these datasets, jargon is being used in a document while in a social media context the jargon is being used informally in a forum of multiple speakers. I can imagine several practical uses for this in industry. Jargon can be used as a way of identifying in-members of a group (Concise Oxford Companion to the English Language). Say an advertiser wants a way to systematically cater their advertisements to a specific community. And even in further applications, it may be possible and a bit morally ambiguous to create a model to determine which communities a user belongs to using their jargon.

Vivaldi et al. (2010) used a Wikipedia corpus for the reason that wikipedia has categories that contain different articles and this structure facilitates creating domain of articles that represents a particular topic. Following similar logic, this project will use reddit as a corpus because the structure of reddit facilitates creating domains on a particular topic. Reddit contains thousands of subreddits, forums where users post and comment about certain topics. In this project, text from a single subreddit represents a single domain. Because reddit contains hundreds of thousands of subreddits (any user can create a subreddit about just any topic), the goal here is to develop a pipeline to computationally extract the domain-specific terms for any community of internet users that has a subreddit.

Subreddit corpus:

Using Pushshift.io, ConvoKit developers at Cornell collected posts and comments from Reddit organized by subreddit. The copyright is a MIT license, which allows for commercial use, modification, distribution, and private use on the condition that the license is included in all copies or substantial portions in the software (https://github.com/CornellNLP/Cornell-Conversational-Analysis-Toolkit/blob/master/LICENSE.md). The corpus is quite large, including a total of 948,169 subreddits that contain all the comments and posts from when each subreddit was created to when the data was collected in October 2018. So in other words the entire thing is far too large to store on a single computer and I imagine the runtime to processing that amount to not be feasible. A smaller collection of subreddits of managable size has also been created, which was used for preliminary results, however I intend to create a larger collection for the final project.

TF-IDF

TF-IDF (Term Frequency, inverse document frequency) will be the unsupervised, statistical method used to determine which terms are specific to the domain of a subreddit. This equation is used often in information retrieval to search for documents given a query, as it determines the relevancy of a document based on a query. Term Frequency represents how many times a term appears in a given document, and Inverse Document Frequency represents how many documents contain that term. So for every word in a query, documents will be ranked higher if the words in the query are relevant in the document by summing the TF-IDF for each term. For this project, however, we will be modifying TF-IDF such that terms will be ranked based on their relevancy in a document, which in this project is an entire subreddit. There are a variety of different ways to calculate both term frequency and inverse document-frequency. The following equation is a simple version of TF-IDF yet was found to be quite effective in preliminary results.



In the above equation,c(t, doc) is the frequency of a term in a document, length(doc) is the length of a document, N is the total amount of documents, and k(t) is the total amount of documents that have term k.\* In these results, the algorithm just printed the terms of the top 100 TF-IDF scores, though this will likely be modified. Kim et al. (2009) used a similar TF-IDF equation and found satisfactory results.

Pos tagging

Before TF-IDF is applied to find relevance scores for each term of the document, a POS tagger will be applied to the corpus. Firstly, often times a patriculard domain-specific term is only different from a word outside the community by a part of speech. For example, if we analyzed a dating subreddit, we would expect the algorithm to detect the verb “ghost” (to break a promise to show up) despite the very common noun “ghost” (spirit, phantom). Secondly, I think it would be important information for a potential user to know.

N-grams

Another important factor of this implementation is the inclusion of bigrams, trigrams, and possibly higher n-grams. Many domain-specific terms are a combination of two or more words. In the preliminary results, the algorithm analyzed r/IndianaUniversity and was able to detect the word “Hutton” but not the full term “Hutton Honors College.” This will likely need to be integrated into the TF-IDF equation somehow. In the preliminary results which only searched for unigrams, the terms detected had scores very close to 4.635 with a very narrow margin and a bigram or trigram will probably output a different score. Especially if the algorithm continues to select the top 100 terms with the largest scores, the difference in score of a bigram from a unigram will need to be considered in the equation.

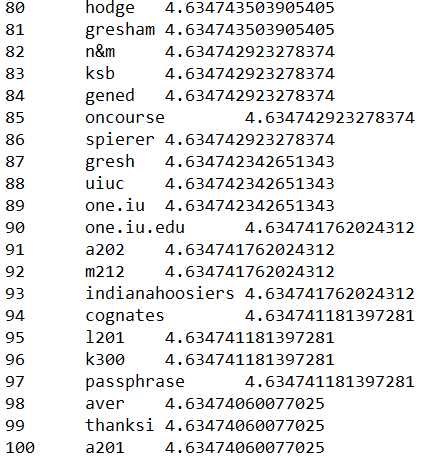
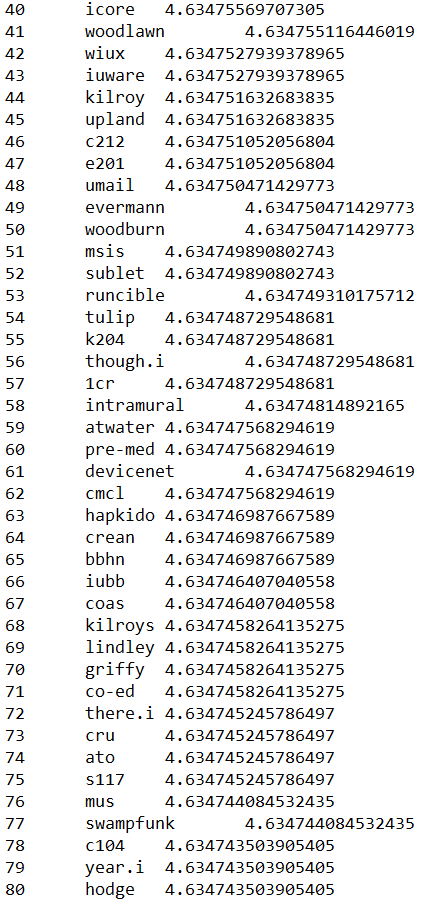
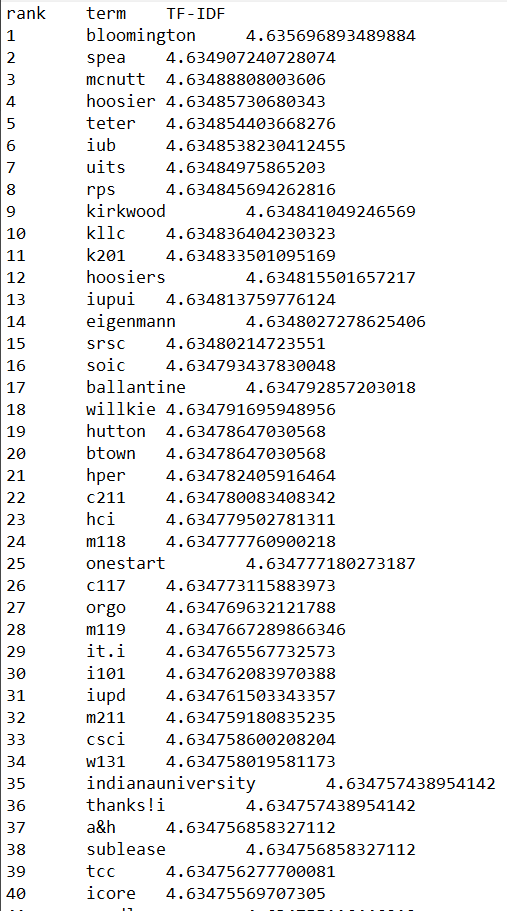
Evaluation:

Unfortunately, a sound method of computationally evaluating the performance of the algorithm does not appear to exist. One could guess at the probable domain-specific terms of a given community but it’s possible that the algorithm could pick up a real domain-specific term that the human failed to consider. Kim et al. 2009, the project that this is most similar to, employed evaluators to go through the results and label each term manually. As I do not have the money to employ anyone, I have the option of doing this myself, however as I am the one programming this algorithm I have a certain bias towards being leniant with the algorithm. Another option I see is to let others, possibly personal friends or classmates, choose a subreddit of their interest and have them label a hundred or so terms detected by the algorithm. The latter seems to be the most reliable option to me, though still has the problem that usually a member of a community does not have knowledge of all the jargon. For example, although I am a member of the IU community, I had to look up the term HPER which led me to “Health, Physical Education, and Recreation”. Unfortunately, there seems to be know way to computationally evaluate the performance.

Preliminary results

A preliminary attempt at this project was fully implemented and preliminary results were achieved. The subreddit r/IndianaUniversity was compared to a downsized subreddit corpus. Cornell offers a small version of the corpus with 100 diverse subreddits cut to include 100 comment threads of at least 10 comments each (cornell). This version was used for the fast processing speed and low memory of a small corpus that is adequate for preliminary results. A larger corpus will be created for the final project. The top hundred results of TF-IDF on unigrams was printed out.

The results look promising at this point. Names of residential and academic buildings (McNutt, Teter, Ballantine, Eigenmann), class numbers (K201, C211, M118, C117), academic programs (SOIC, CSCI, SPEA), and on-campus organizations (Hutton, RPS, IUPD, UITS) frequent the top results. However, terms such as “sublease” and “sublet” both appear in the top hundred results and are not domain-specific terms for the Indiana University community. I expect this is due to a subreddit for a university being compared to subreddits that are for universities. As I imagine all university subreddits contain comments about discussing subleasing, I expect the solution for this case and cases like it is a larger and more diverse corpus.



Long term and short term goals:

So from the words extracted by the algorithm, we can see that the goal of this project, to develop a pipeline to find the jargon words of any community of people that have a subreddit, is nearly accomplished. There are still random words that should be weeded out, but any student at Indiana University will know most of these terms and most terms are specific to Indiana University. There are plenty of ways to improve the above algorithm above by implementing POS tagging and the use of N-grams, or changing the cutoff point to something other than selecting the top hundred words by TF-IDF, but these are just improvements to a pipeline that’s already rather functional. (This is also the connection to the class, as n-grams, POS tagging, and word frequencies were all topics that were covered). These are the short term goals of this project. This leaves me at a loss of a long term goal for this project that goes beyond the scope of this project. I mentioned this in the introduction but perhaps this project can be to use the data from this algorithm categorizing users of other social media based on which communities they belong by only using their text as input.

Citations/related works

S. Kim, T. Baldwin, and M. Kan. An unsupervised Approach to Domain-Specific Term Extraction. In *Proceedings of the Australasian Language Technology Association Workshop 2009.* Pages 94-98. 2009. url: https://www.aclweb.org/anthology/U09-1013

S. Kim and L. Cavedon. Classifying Domain-Specific Terms Using a Dictionary. 2011. In *Proceedings of the Australasian Language Technology Association Workshop 2011*. Pages 57-65. url. <https://www.aclweb.org/anthology/U11-1009.pdf>

T. McArthur. Jargon. *Concise Oxford Companion to the English Language.* Retrieved October 16, 2020 from Encyclopedia.com: https://www.encyclopedia.com/humanities/encyclopedias-almanacs-transcripts-and-maps/jargon

A. Meyes, Z. Glass, A. Greive-Smith, Y. He, S. Liao, and R. Grishman. 2014. Jargon-Term Extraction by Chunking. In *Proceedings of the COLING Workshop on Synchronic and Diachronic Approaches to Analyzing Technical Language*. Pages 11-20. url: <https://www.aclweb.org/anthology/W14-6002.pdf>

Z. Pei, Z. Sun, and Y. Xu. Slang Detection and Identifiction*.* In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL).* 2019*.* Pages 881-889. url: <https://www.aclweb.org/anthology/K19-1082.pdf>

J. Vivaldi and H. Rodriguez. Finding Domain Terms using Wikipedia. In *Proceedings of the International Conference on Language Resources and Evaluation*. 2010. Pages 17-23. url: https://www.researchgate.net/publication/220746514\_Finding\_Domain\_Terms\_using\_Wikipedia

L. Weisong and S. Cai. Translating Electronic Health Record Notes from English to Spanish: A Preliminary Study*. In Proceedings of BioNLP 15*. 2015. Pages 134-140. url: <https://www.aclweb.org/anthology/W15-3816.pdf>

\*Not really sure how to cite this but I’m currently in an information retrieval course (ILS-Z534) and all the information relating to TF-IDF comes from my professor’s slides and other class content.