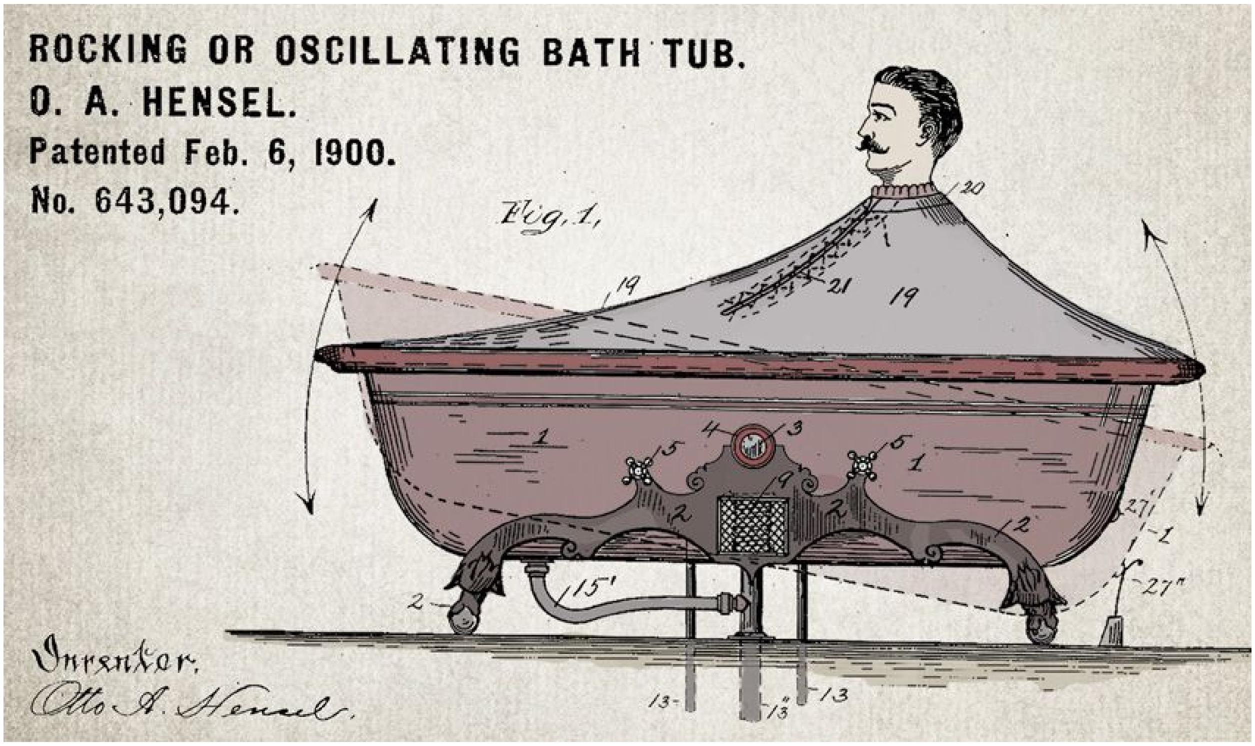
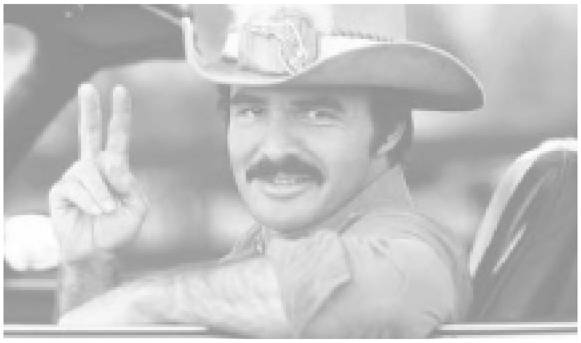
Establishing a Baseline Search Method for USPTO Patent Claim Information

Information Retrieval and Natural Language Processing using Python



By Joseph Conran
For Natural Language Processing, Dr. Steven Kunath

Submitted December 14, 2018



In Memory of Burt Reynolds

Abstract

Inspired by a request for information submitted by the US Patent and Trademark Office (USPTO), interest in creating a search engine for patent claims was realized through the implementation of two models using Python. Patent claim data was downloaded from the USPTO Research Database and subsetted. This information was then indexed using multiple syntactical language processes. The search model processes the search query similarly to the index and then matches the query to the index using two different models. The first model implements cosine similarity and the second model uses term frequency–inverse document frequency (TF-IDF). Both models rank order the search index by the algorithm result and select the top matching patent. The models then call upon the PatentsView annotated program interface (API) to return relevant information about the prosecution of the patent and and also print out the full text of the patent.

The USPTO Patent Claim Search Engine

The US Patent and Trademark Office (USPTO) submitted September 13, 2018 a Request for Information (RFI)[1] "to conduct market research and seek expertise in Artificial Intelligence (AI) capabilities as they specifically relate to 'search' during patent application prosecution and to assess vendor/organization capabilities and interest." This RFI sought answers to five different topics which include:

1. Expanding the knowledge corpus beyond patents
2. Algorithmic classification of knowledge corpus
3. Machine retrieval of representative prior art within corpus
4. Summarization/knowledge extraction of retrieved information
5. Capturing search "journey" and identifying when it is complete

While the scope of this RFI is beyond perhaps an initial foray into natural language processing, it provides a specific and relevant real-world problem to solve, especially focusing on algorithmic classification of the knowledge corpus, machine retrieval of information, and extraction of retrieved information. In essence, the authors of this paper seek to build a search engine.

USPTO and the Patent Claim Dataset

The United States Patent and Trademark Office is an agency of the US Department of Commerce that issues patents for inventions and trademarks for registration for intellectual property identification. Since 1790, 10 million US patents have been issued with about 150,000 issued every year in the modern era. It is imperative then for both patent examiner, prosecutors, and businesses filing to be able to sort through the relevant patents and their claim text to ascertain whether new patents can be correctly registered.

For the purpose of building a search engine, USPTO's "Patent Claims Research Dataset"[2] was downloaded in a compressed .csv format and extracted. The raw dataset (29 gigabytes) consists of 110,354 total patents with 1,048,575 total claims. The dataset includes patent number, claim number, claim text, dependent ID, independent flag, and application ID. A sample of the dataset is seen in Appendix Figure 1. To save on computational resources, the last 10462 rows were subsetted which amounts to 1088 patents with 10462 claims.

Natural Language Processing and Information Retrieval

[1] *Govtribe.com*, govtribe.com/opportunity/federal-contract-opportunity/uspto-s-challenge-to-improve-patent-search-with-artificial-intelligence-rfiusptoaipatentseach18#.

[2] Office of Chief Economist. "Patent Claims Research Dataset." *United States Patent and Trademark Office - An Agency of the Department of Commerce*, www.uspto.gov/learning-and-resources/electronic-data-products/patent-claims-research-dataset.

Natural language processing is a field of computer science that focuses on how computers process natural (formed without conscious planning) language data. The primary objectives of NLP focus on syntax, semantics, discourse, and speech.[3] Information retrieval or IR is the computer retrieval of relevant information from a collection of information resources.[4]

An information retrieval search engine requires two components: an index and a searching algorithm. To create the index, each (or whatever information) must be assigned search IDs and then syntactically processed and weighted. The search engine then takes a user query, processes the query's syntax similarly to the index, finds relevant documents based off an algorithm, ranks the documents, and returns a selection to the user.

Patent Claim Search Engine Setup

The intent of this analysis is to create a search engine capable of taking a technical search query and return relevant patent metadata and the full text of the patent claim. The methodology and raw code will outline the full process of creating this search engine. An outline of the steps includes pulling in the USPTO patent ID, claim number and claim text to create a search ID and then processing the natural language claim text into a machine readable index. The engine then takes a search query and processes it similarly to the index, compares the processed search term to the index using “term frequency-inverse document frequency” (TF-IDF) or cosine similarity algorithms, ranks the documents by the algorithm score, selects the top matching patent claim, and uses an application program interface (API) to retrieve the relevant patent metadata and full text of the patent.

Cosine Similarity

A general mathematical function used to measure the cosine angle of two product vectors, this formula is used in natural language processing to compare the similarity of two text strings.[5] Its main difference over TF-IDF is that it does not rely on the frequency of terms, but rather the commonality of terms between the two strings. Wikipedia provides the mathematical function as follows:

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2} \sqrt{\sum\limits_{i=1}^{n} B_i^2}},$$

where $\mathbf{A_i}$ and $\mathbf{B_i}$ are components of vectors **A** and **B**.

[3] “Major Evaluations and Tasks.” *Wikipedia*, Wikimedia Foundation, 9 Dec. 2018, en.wikipedia.org/wiki/Natural_language_processing#Major_evaluations_and_tasks.

[4] Jansen, Bernard J., and Soo Young Rieh. “The Seventeen Theoretical Constructs of Information Searching and Information Retrieval.” *Journal of the American Society for Information Science and Technology*, 2010, doi:10.1002/asi.21358.

[5] “Cosine Similarity.” *Wikipedia*, Wikimedia Foundation, 10 Dec. 2018, en.wikipedia.org/wiki/Cosine_similarity.

TF-IDF

Term Frequency-Inverse Document Frequency (TF-IDF) is a statistical metric that attempts to quantify the importance of a word to a document in a corpus. The theory behind TF-IDF relies on two basic concepts:

1) A term that appears many times within a document is likely to be more important than a term that appears only once, and
2) A term that occurs in a few documents is likely to be a better discriminator than a term that appears in most or all documents.

TF-IDF creates a metric that increases as the number of occurrences of the term increases in the document but decreases as the number of documents in which the term occurs increases.[6] TF-IDF is the consists of two statistics: term frequency and inverse document frequency. The metrics are calculated as:

Term Frequency
term t in document d

$$f_{t,d} \Big/ \sum_{t' \in d} f_{t',d}$$

Inverse Document Frequency
N = total docs in corpus
nt = total docs with term t

$$\log \frac{N}{n_t}$$

Inverse document frequency is always above 1, which would indicate that the term occurs in all documents and thus has no importance to document ranking. This is achieved by taking the logarithm of the IDF, as *log(1) = 0*.[7]

The product of the term frequency and the inverse term frequency are then multiplied to get the final TF-IDF metric. A high metric value for a term is achieved for if that term has a high document frequency and a low corpus frequency and indicates higher importance to the document. This filters out common terms that have less importance as these terms will have lower IDF scores. Documents are converted to vectors with values being each term's TF-IDF score. Documents and queries can be compared and ranked through the computing of the cosine similarity between the various TF-IDF vectors.

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

$tf_{i,j}$ = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents

Implementation[8]

Cosine Similarity

[6] *Document Similarity in Information Retrieval*, courses.cs.washington.edu/courses/cse573/12sp/.

[7] "Term Frequency - Inverse Document Frequency Statistics." *TFIDF Statistics | SAX-VSM*, jmotif.github.io/sax-vsm_site/morea/algorithm/TFIDF.html.

[8] Please see attached .pdf or .ipynb for documented Python Notebook code

To build the search engine using cosine similarity, first the patent claim raw data subset was loaded into Python as a .csv file using the Jupyter Notebook IDE. The patents claims were then concatenated into one patent claim row by their patent ID. Each patent full text was then processed using Python and the NLTK package as follows:

1. Remove Digits
2. Remove Special Characters
3. Lowercase
4. Tokenize, or break text string into sequence of words
5. Remove stop words, or no-value words (the, who, an, etc)
6. Stemming, or reducing a word to base or root stem (Snowball stemmer)

Once the patent claim text was processed, every patent, by patent number, was indexed by the 30 most common words from the processed text. While arbitrary, thirty words notionally seemed to capture the most relevant information without extending too far into the string of text.

The input for a query search term expects a certain level of specificity or technicality over a natural language question—Inputting "liquid crystal display" versus "provide me with documents containing the words liquid crystal display". While preprocessing to remove stop words and natural language may reduce some of the confounding search terms, best practices will improve the search results.

Once the search terms are inputted, they are processed similarly to the indexed keywords with an added step of autocorrecting any spelling mistakes. This step is not necessarily additive but has been included as a proof of concept. The processed query is then compared against the index of the top 30 keywords using the cosine similarity function and assigned a numerical score. The search index is then sorted in descending values and the top matching patent number is selected.

Taking the top matching patent number, the patent metadata including: Assignee, Country, Key ID, Organization, Patent Examiner, ID, Last Name, Inventor, Inventor ID, Inventor Last Name, Patent Number, and Patent Title are requested in JSON (JavaScript Object Notation) format using the third-party PatentsView API. From the index dataframe built previously the matching full text of the patent claim is also returned.

Term Frequency – Inverse Document Frequency

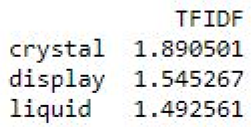
To build the search engine using TF-IDF, the sample patent dataset was similarly loaded and parsed as the Cosine Similarity method. The patent text was segmented with the Punkt Sentence Tokenizer and tokenized with the Treebank Word Tokenizer. The tokenized text was then preprocessed, with claims numbers, stop words, and special characters being removed and the remaining text being converted to lowercase and stemmed with the Porter Stemmer.

To calculate the TF-IDF score for each term in each document, a corpus-wide list of unique terms was created along with a reference dictionary with keys being document ID's and values being the tokenized text of the document. The term frequency vector for each document was calculated by counting the number of occurrences of each term from the corpus list in the tokenized document text. The inverse document frequency for each term was calculated by iterating over each document in the corpus and if that term occured in the document's tokenized text, the document count would increment by 1. The total corpus length was divided by the final document count and the log of this number was taken to calculate the IDF for each term in the corpus. The IDF score for each was term multiplied by each term frequency count for that term in each document to calculate the final TF-IDF scores.

The process of parsing and vectorizing the query followed a similar path as the patent documents. The cosine similarity between the query TF-IDF vector and each document TF-IDF vector was calculated, all results with a zero TF-IDF were removed, and the remaining documents were sorted in descending order. The documents with the ten highest cosine similarities to the query are extracted to produce the final rank of relevant documents.

Analysis

TF-IDF



```
            TFIDF
crystal  1.890501
display  1.545267
liquid   1.492561
```

An experimental query of "liquid crystal display" was passed the parser and TF-IDF calculator to test the system. Removing all documents with a zero TF-IDF successfully resulted in all relevant documents containing at least one of the terms in the search query. The top 2 documents (patents 4040719 and 4040720) both contained all three query terms. Patent 4040719 contained more occurrences of "crystal", while patent 4040720 contained more occurrences of "display". As expected by the query term TF-IDF scores as seen above, "crystal" has more weight than "display" due to their respective corpus rarities. However, as seen in Figure 3, there are some odd outputs to the model. Higher values of single terms counts result in a worse TF-IDF score. In addition, patent 4040721, which contains more occurrences of each term than any other patent, is ranked 6th in terms of TF-IDF score. This anomaly is possibly a result of the cosine similarity punishing document length in the denominator of the metric. TF-IDF also does not take into account the theory that matching all words of a query most likely implies more document relevance than matching a single term regardless of term frequency.

Another ranking system using the TF-IDF vectors was tested to compare results to cosine similarity. Rather than divide the vector dot product of the query and document by the lengths of the vectors, the dot product itself was used as the similarity metric. Next, the total number of query terms matched in the document were counted. The documents were finally ordered first by number of matching terms and then by dot product scores. The results of this analysis are seen in Figure 3. In this analysis, patent 4040721 is determined to be the most relevant document, followed by patents 4040719 and 4040720, the most relevant documents from the cosine similarity measure. Though less relevant documents have higher dot product scores than more relevant documents, this measure takes into account the theory that more matching words means more relevance.

Finally, a combination of cosine similarity and matching terms was tested. This measure takes into account both number of query terms matched and length of the documents, which acts on the theory that term frequency in relation to document length is more telling than simple term frequency. The results, as seen in Figure 3, follow this theory, with relevance sorted by number of terms matched and the ratio of matched terms to document lengths.

Cosine Similarity

Reviewing the keyword index, the concatenated claims are correctly processed and then indexed by the 30 most frequent keywords (see Appendix Figure 4). Search term processing also occurs correctly with an extreme example of the search input “LiQUId CrrYstal Displayying $1” processed as ['liquid', 'crystal', 'display']. The search input is then compared against each row of keywords with a cosine similarity value properly appended to the dataframe. For additional testing, the search term was also compared against non-weighted processed patents prior filtering by the top 30 keywords. These values were then properly sorted by the top ranking match as seen in Appendix Figure 5. Finally, the top matching patent claim number by cosine similarity against processed keywords is properly pulled from the ranked search index and entered into the PatentsView API. The API returns the desired JSON information and the full text of the patent is outputted as seen in Figure 5.

Further Research

While beyond the scope of this implementation, it should be noted that the conceptual models for TF-IDF and cosine similarity do return the same top result for the search query “liquid crystal display” with small differences in the other top results. Cosine similarity on the non-sorted processed claim text produced a different and notionally less accurate result for the search query as the top result matched to a patent on heating a liquid and not a display made of “liquid crystal.” As two baseline models have been developed, sensitivity testing and user feedback can now be implemented to refine the model and improve the search results.

Other opportunities for research include improving the efficiency of the code, expanding the dataset, using other searching algorithms such as n-gram similarity, regex matching, or fuzzy matching. Machine and deep learning, as the original USPTO RFI desired, could also be implemented as search functions.

Conclusion

These baseline models are shown to be capable of extracting and sorting relevant documents related to a query. Cosine similarity and TF-IDF both produced similar results that returned documents containing some amount of the query terms, with top documents containing all query terms, as is needed for a successful search engine. However, through manual examination, it was determined that both measures returned rankings that were slightly out of order of logical relevance.

This demonstrates the reality that there is no easy or universal solution to finding relevant documents and a solution must be tailored to the specific problem. As shown in the results, simple cosine similarity and TF-IDF scores value the importance of how many of the individual query terms are in the document terms less than the overall query term frequency. The number of query terms found in the document must be weighted along with cosine similarity and TF-IDF scores to return relevance ordering satisfactory for a search engine. The cosine similarity of TF-IDF vectors with numbers of terms matches produces the most logical relevance ranking. Through further research and user surveys, this engine can be improved and made more efficient.

References

Jansen, Bernard J., and Soo Young Rieh. “The Seventeen Theoretical Constructs of Information Searching and Information Retrieval.” *Journal of the American Society for Information Science and Technology*, 2010, doi:10.1002/asi.21358.

Document Similarity in Information Retrieval, courses.cs.washington.edu/courses/cse573/12sp/.

Govtribe.com, govtribe.com/opportunity/federal-contract-opportunity/uspto-s-challenge-to-improve-patent-search-with-artificial-intelligence-rfiusptoaipatentseach18#.

Jansen, Bernard J., and Soo Young Rieh. “The Seventeen Theoretical Constructs of Information Searching and Information Retrieval.” *Journal of the American Society for Information Science and Technology*, 2010, doi:10.1002/asi.21358.

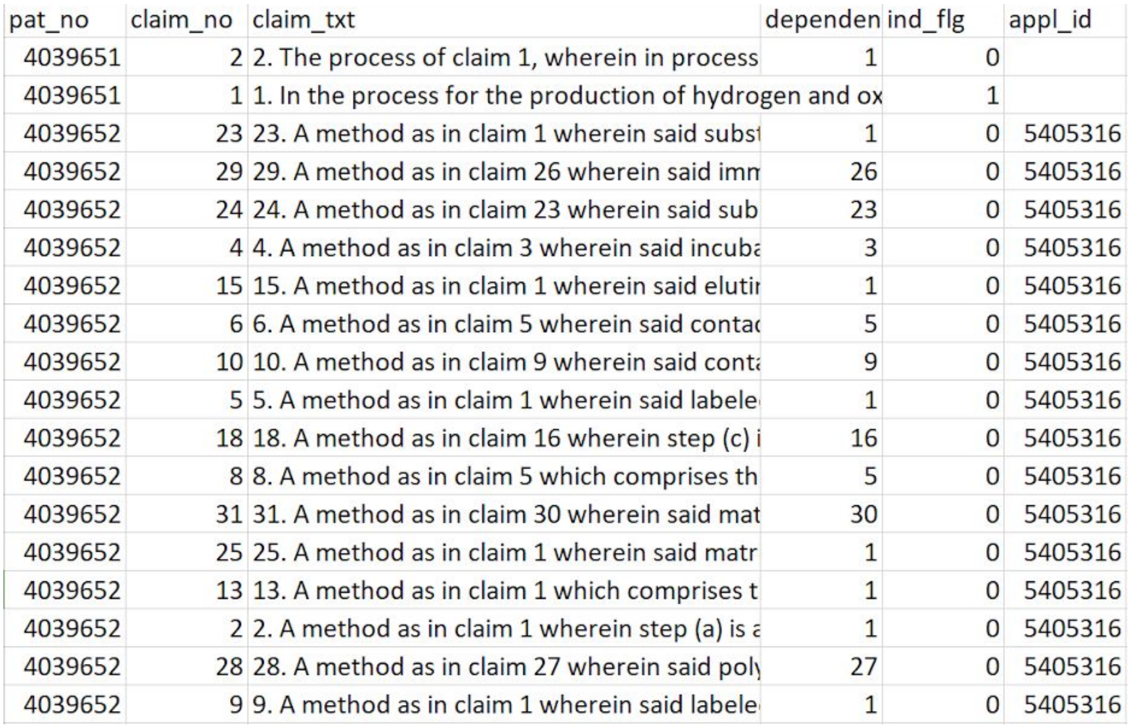
Major Evaluations and Tasks.” *Wikipedia*, Wikimedia Foundation, 9 Dec. 2018, en.wikipedia.org/wiki/Natural_language_processing#Major_evaluations_and_tasks.

Office of Chief Economist. “Patent Claims Research Dataset.” *United States Patent and Trademark Office - An Agency of the Department of Commerce*, www.uspto.gov/learning-and-resources/electronic-data-products/patent-claims-research-dataset.

“Term Frequency - Inverse Document Frequency Statistics.” *TFIDF Statistics | SAX-VSM*, jmotif.github.io/sax-vsm_site/morea/algorithm/TFIDF.html.

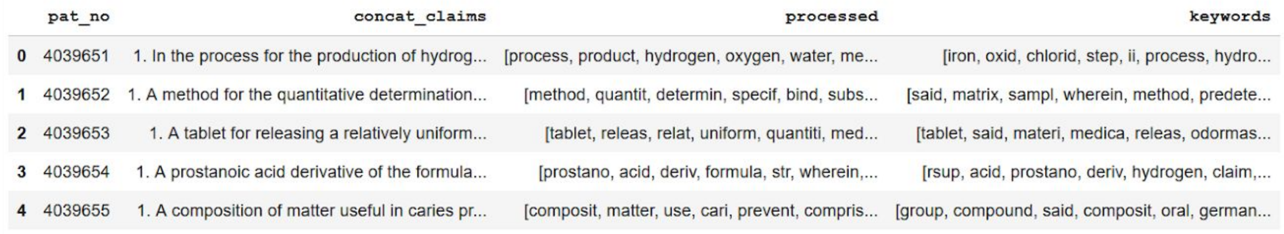
Appendix

Figure 1



pat_no	claim_no	claim_txt	dependen	ind_flg	appl_id
4039651	2	2. The process of claim 1, wherein in process	1	0	
4039651	1	1. In the process for the production of hydrogen and ox		1	
4039652	23	23. A method as in claim 1 wherein said subst	1	0	5405316
4039652	29	29. A method as in claim 26 wherein said imm	26	0	5405316
4039652	24	24. A method as in claim 23 wherein said sub	23	0	5405316
4039652	4	4. A method as in claim 3 wherein said incuba	3	0	5405316
4039652	15	15. A method as in claim 1 wherein said elutir	1	0	5405316
4039652	6	6. A method as in claim 5 wherein said contac	5	0	5405316
4039652	10	10. A method as in claim 9 wherein said conta	9	0	5405316
4039652	5	5. A method as in claim 1 wherein said labele	1	0	5405316
4039652	18	18. A method as in claim 16 wherein step (c) i	16	0	5405316
4039652	8	8. A method as in claim 5 which comprises th	5	0	5405316
4039652	31	31. A method as in claim 30 wherein said mat	30	0	5405316
4039652	25	25. A method as in claim 1 wherein said matr	1	0	5405316
4039652	13	13. A method as in claim 1 which comprises t	1	0	5405316
4039652	2	2. A method as in claim 1 wherein step (a) is a	1	0	5405316
4039652	28	28. A method as in claim 27 wherein said poly	27	0	5405316
4039652	9	9. A method as in claim 1 wherein said labele	1	0	5405316

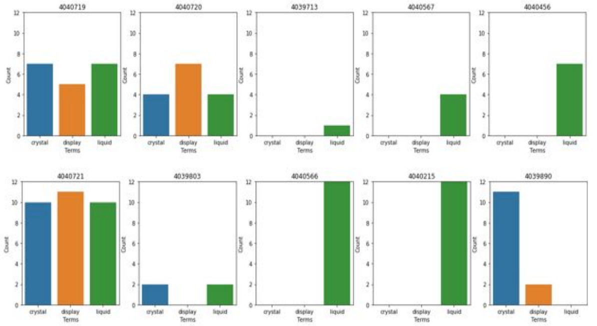
Figure 2

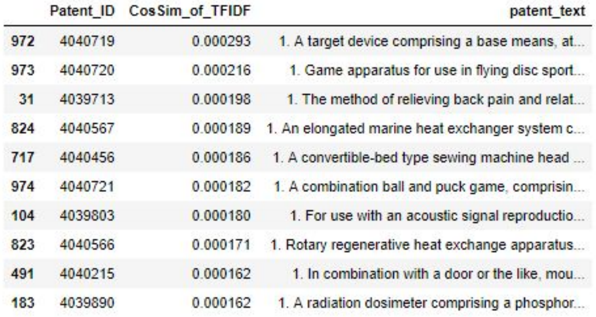


	pat_no	concat_claims	processed	keywords
0	4039651	1. In the process for the production of hydrog...	[process, product, hydrogen, oxygen, water, me...	[iron, oxid, chlorid, step, ii, process, hydro...
1	4039652	1. A method for the quantitative determination...	[method, quantit, determin, specif, bind, subs...	[said, matrix, sampl, wherein, method, predete...
2	4039653	1. A tablet for releasing a relatively uniform...	[tablet, releas, relat, uniform, quantiti, med...	[tablet, said, materi, medica, releas, odormas...
3	4039654	1. A prostanoic acid derivative of the formula...	[prostano, acid, deriv, formula, str, wherein,...	[rsup, acid, prostano, deriv, hydrogen, claim,...
4	4039655	1. A composition of matter useful in caries pr...	[composit, matter, use, cari, prevent, compris...	[group, compound, said, composit, oral, german...

Figure 3

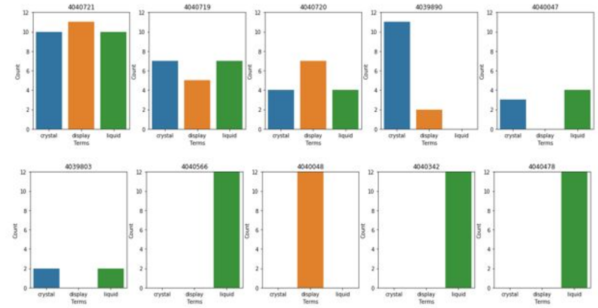
Cosine Similarity Counts

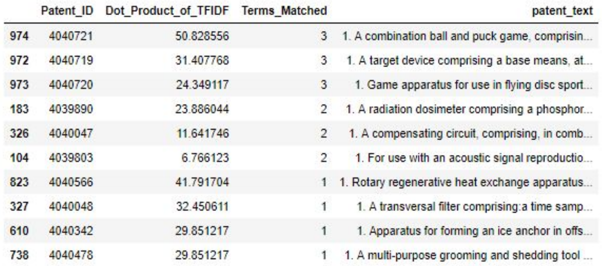




	Patent_ID	CosSim_of_TFIDF	patent_text
972	4040719	0.000293	1. A target device comprising a base means, at...
973	4040720	0.000216	1. Game apparatus for use in flying disc sport...
31	4039713	0.000198	1. The method of relieving back pain and relat...
824	4040567	0.000189	1. An elongated marine heat exchanger system c...
717	4040456	0.000186	1. A convertible-bed type sewing machine head ...
974	4040721	0.000182	1. A combination ball and puck game, comprisin...
104	4039803	0.000180	1. For use with an acoustic signal reproductio...
823	4040566	0.000171	1. Rotary regenerative heat exchange apparatus...
491	4040215	0.000162	1. In combination with a door or the like, mou...
183	4039890	0.000162	1. A radiation dosimeter comprising a phosphor...

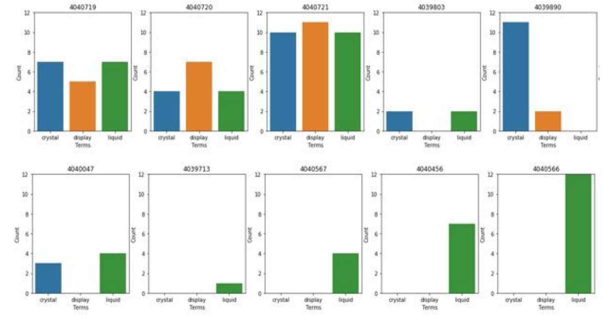
Dot Product with Matching Term Counts

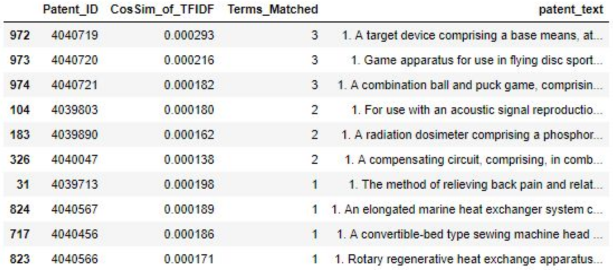




	Patent_ID	Dot_Product_of_TFIDF	Terms_Matched	patent_text
974	4040721	50.828556	3	1. A combination ball and puck game, comprisin...
972	4040719	31.407768	3	1. A target device comprising a base means, at...
973	4040720	24.349117	3	1. Game apparatus for use in flying disc sport...
183	4039890	23.886044	2	1. A radiation dosimeter comprising a phosphor...
326	4040047	11.641746	2	1. A compensating circuit, comprising, in comb...
104	4039803	6.766123	2	1. For use with an acoustic signal reproductio...
823	4040566	41.791704	1	1. Rotary regenerative heat exchange apparatus...
327	4040048	32.450611	1	1. A transversal filter comprising:a time samp...
610	4040342	29.851217	1	1. Apparatus for forming an ice anchor in offs...
738	4040478	29.851217	1	1. A multi-purpose grooming and shedding tool ...

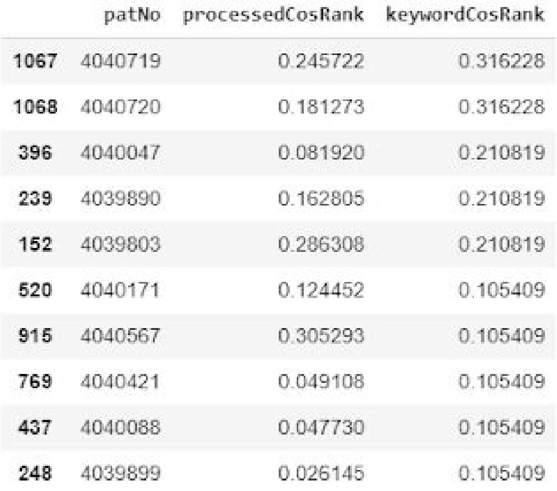
Cosine Similarity with Term Frequency





	Patent_ID	CosSim_of_TFIDF	Terms_Matched	patent_text
972	4040719	0.000293	3	1. A target device comprising a base means, at...
973	4040720	0.000216	3	1. Game apparatus for use in flying disc sport...
974	4040721	0.000182	3	1. A combination ball and puck game, comprisin...
104	4039803	0.000180	2	1. For use with an acoustic signal reproductio...
183	4039890	0.000162	2	1. A radiation dosimeter comprising a phosphor...
326	4040047	0.000138	2	1. A compensating circuit, comprising, in comb...
31	4039713	0.000198	1	1. The method of relieving back pain and relat...
824	4040567	0.000189	1	1. An elongated marine heat exchanger system c...
717	4040456	0.000186	1	1. A convertible-bed type sewing machine head ...
823	4040566	0.000171	1	1. Rotary regenerative heat exchange apparatus...

Figure 4



	patNo	processedCosRank	keywordCosRank
1067	4040719	0.245722	0.316228
1068	4040720	0.181273	0.316228
396	4040047	0.081920	0.210819
239	4039890	0.162805	0.210819
152	4039803	0.286308	0.210819
520	4040171	0.124452	0.105409
915	4040567	0.305293	0.105409
769	4040421	0.049108	0.105409
437	4040088	0.047730	0.105409
248	4039899	0.026145	0.105409

Figure 5

"count": 1,
"patents": [
"assignees": [
"assignee_country": null,
"assignee_key_id": null,
"assignee_organization": null
"examiners": [
"examiner_id": "87vxycv5jpvf3o4l0wam0g34z",
"examiner_last_name": "Corbin"
"examiner_id": "ljfgrp5jxfn12mag4p4rjp5da",
"examiner_last_name": "Hille"
"inventors": [
"inventor_first_name": "Heinz F.",
"inventor_key_id": "124686",
"inventor_last_name": "Schiebelhuth"
"patent_number": "4040719",
"patent_title": "Frequency indicator for receiving devices"
"total_patent_count": 1

```
Full Text:1067    1. A display panel having a plurality of liqui...
Name: concat_claims, dtype: object
```

Google Drive Link for Documents
https://drive.google.com/open?id=1m7nDiczLt5JwcTf_Za07Qc2fTSotaItX