Master 1 in Computer Science

Project report

Semantic Causality for a corpus of documents

Authors:
Agliz Yasmine and Haddad Ayale

Supervisor:
Affeldt Séverine

Academic year 2018-2019

Contents

List of Figures

1

Introduction

1.1 Context and motivations

MEDLINE is a biomedical literature database commonly used to access information essential for research and health care. Life science researchers usually rely on MEDLINE to keep up with the latest development in the medical field. Recently, there has been an exponential rate of growth of the biomedical literature. With the increasing volume of publications, it became critical to efficiently access and analyze text information.

Natural language processing is currently at the center of the computer science and linguistics disciplines. It addresses the computational aspects of automatic text processing. This field provides a fertile ground for the algorithms of machine learning. The challenges presented when processing natural language offer new opportunities to the existing machine learning methods and promote the development of new ones.

The representation of words by vectors facilitates the use of unstructured data such as text data. According to the distributional hypothesis, words that appear in similar contexts have close significance. Moreover, words with similar meaning have similar vectors or at least vectors with the same orientation. Thus, the representation of words by vectors allows us to determine the link between words.

There are many branches and research groups working on word embeddings such as Google and Facebook. The word vector matrix can be used as a support for the automatic discovery of relationships between words. In this project, we are particularly interested in causal links. The identification of these links is possible with the use of some approaches such as the reconstruction of causal networks.

1.2 Contributions and organization of the report

This work focuses on extracting relevant information from a biomedical corpus. The project was mentored by Madame Affeldt Séverine. The group is composed by two students

from M1- Machine Learning and Data Science: Agliz Yasmine and Haddad Ayale.

This project aims to analyze semantic causality in a corpus of biomedical documents. The First step of the project is to extract words which are related to a certain topic. For this, we will use some co-clustering algorithms. Then the second step is to determine the most significant words. We used two different approaches to get these words. The first one is the frequency of the word then we used another metric which is the pointwise mutual information.
Afterwards, we generate graphs to illustrate the links between the significant words. Finally, we generate directed graphs in order to add the causal information between the words. At the end of this project we will discuss the results of the different approaches.

2

Used tools and technologies

2.1 Corpus

The corpus of documents used in this project is a set of Pubmed's abstracts. Pubmed is a free search engine for bibliographic data from all areas of specialization in biology and medicine. It is the free version of Medeline, a bibliographic database in biomedical sciences which contains more than 30 million biomedical references. Pubmed10 is a corpus of 10 different diseases. For this projet, we have decided to pick only 5 diseases in the first place. We chose the five most distant diseases to guarantee good results for the clustering algorithm. The diseases are: Hay fever, Kidney calculi, Age-related Macular Degeneration, Migraine, Otitis.

Pubmed10 data set is a collection of 15,500 medical documents, partitioned across 10 different diseases. It consists of published abstracts in the MEDLINE database from 2000 to 2008. There is already an adjacency matrix $\{word, document\}$ available online. We chose to use this matrix instead of creating one ourselves. This would allow us to compare our results to the ones on the internet.

2.2 Co-clustering methods

The abstracts in Pubmed10 are not classified according to the disease. To obtain a separation of abstracts according to the disease, we applied a co-clustering algorithm to the adjacency matrix. The co-clustering[1] algorithm is a technique which aims to perform a simultaneous partition of the rows and columns of a data matrix. We tried three types of co-clustering in this project: coClustMod, coClustSpecMod, coClustInfo.

The two algorithms CoClustMod and CoclustSpecMod, seek an optimal block-diagonal clustering, meaning that objects and features have the same number of clusters. Then, after proper permutation of the rows and columns, the algorithm produces as result a block-diagonal matrix (see Figure 3). CoClustInfo is a non-diagonal algorithm in the context of document-term matrix.

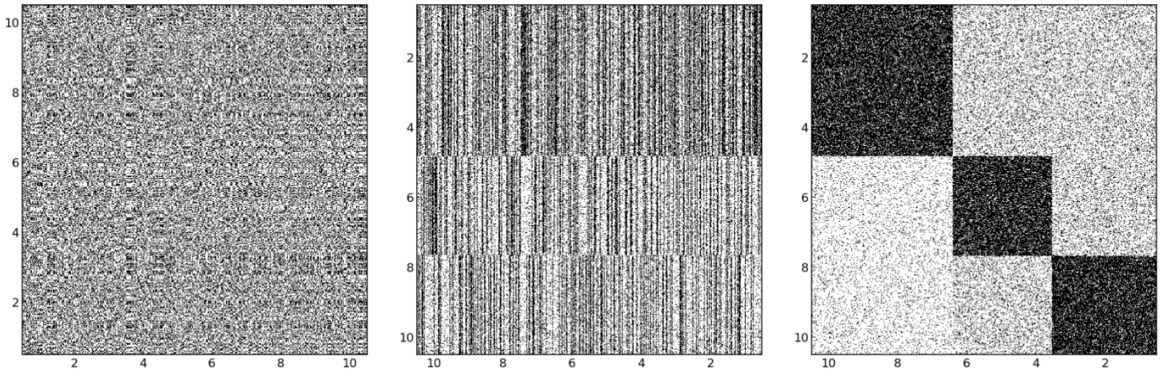

Figure 1: Co-clustering steps: original data(left). Data reorganized according to row clusters(middle). Data reorganized according to row and column clusters(right)

Among the existing approaches, there is CoClustMod which is efficient for large data processing. It also creates a partition of several variables, continuous or nominal. We use this approach as a benchmark throughout this project.

2.3 Embeddings

In order to quantify a semantic relation between two words, we must convert them into mathematic elements. This allows us to compute the similarity between the words. Embeddings are relatively low-dimensional space into which we can translate high dimensional vectors. In this case, it is a mapping from discrete objects, such as words, to vectors of real numbers. There are three known and commonly used tools for word embeddings: Word2Vec[2],

Glove [3] and FastText [4].

2.3.1 Word2Vec

Word2Vec was implemented by a team of researchers at Google. It consists of a two layer neural network which tries to learn a model of words end their corresponding vectors. Word2Vec uses one of the two following architectures to produce distributed word vectors:

Continuous bag-of-words (CBOW): Using the surrounding context words, the model predicts the vector of the current word.
Continuous skip-gram: This architecture also uses the surrounding window of context words. However, it weighs nearby context words more heavily than more distant context words.

CBOW is faster while Skip-gram is slower yet, Skip-gram is more efficient for infrequent words.

2.3.2 Glove

Glove was developed as an open-source project at Stanford. The algorithm is derived from algebraic methods. It consists of factorizing a matrix of words co-occurrence statistics. The algorithm performs very well and converges faster than Word2Vec.
The GloVe model is trained on the non-zero entries of a global $\{word, word\}$ co-occurrence matrix. The matrix tabulates how frequently words co-occur with one another in a given corpus. Populating this matrix requires a single pass through the entire corpus to collect the statistics. For large corpora, this pass can be computationally expensive, but it is a one-time up-front cost.

2.3.3 FastText

FastText is a library developed by Facebook that serves two main purposes, learning of word vectors and text classification. It is based on a neural network for word embedding. FastText brings something innovative to the table, compared to the other algorithms. Rather than considering words being independent of one another, it considers all character sub sequences, within a length range, when computing a representation for a word. Therefore, it provides more accurate word vectors for rare words.

In general, these algorithms are used to create word embeddings of different dimensions using

a certain corpus. In this project, we chose to download word embeddings which are already available on the internet. These embeddings are 300-dimensional vectors. The embeddings from the internet have been learned from corpora such as Wikipedia or Giga Word , two large corpora.

Unfortunately, these embeddings do not provide enough biomedical information. Therefore, we added biomedical embeddings which have been learned on PubMed, PubMed Central and Wikipedia by word2Vec. These embeddings are 200-dimensional vectors.

We used four sets of embeddings:

Dimension	**Window**	**Corpus**	**Vocabulary size**	**Algorithm**	**Lemmatization**
300	5	English Wikipedia dump of February 2017	273992	Word2Vec Continuous Skipgram	True
300	5	Gigaword 5^{th} edition	262269	Global vectors	True
300	5	English wikipedia dump of Febbruary 2017 gigaword 5th edition	260073	FastText Skipgram	True
200	5	Pubmed, Pubmed central and english Wikipedia	5443656	Word2Vec	False

These embeddings will be stored in a structure called Keyedvectors. Keyedvectors is structure implemented by Gensim which is a Python software library of topic modelling. This structure is essentially a mapping between entities and vectors. Each entity is identified by its

string. Some of the perks of using KeyedVectors are their size, they are small objects and need less RAM. They are also fast to load. And most importantly, vectors exported by the Facebook and the Google tools can be load into Keyed Vectors.

2.4 Word similarities

2.4.1 Cosine similarity

One of the commonly used methods to quantify the similarity between two words is the cosine similarity[5]. The cosine similarity allows us to quantify the similarity between two vectors with n dimensions, each vector represents a certain word. This similarity is determined by the cosine of the angle between them.

Let two vectors A and B represent two words, the angle σ is obtained by the scalar product and the norm of the vectors: $\cos\sigma = \frac{A \cdot B}{\|A\|\|B\|}$.

The value $\cos\sigma$ is included in the interval $[-1, 1]$. The value -1 indicates opposite vectors so opposite meanings. Then, the value 1 indicates collinear vectors which means that the words are similar.

The value 0 indicates orthogonal vectors which means independent meanings. Finally, the intermediate values are used to evaluate the degree of similarity.

2.4.2 NPMI

Another way of quantifying the similarity between two words is the NPMI [6] which stands for Normalized Pointwise Mutual Information. Pointwise mutual information is a measure of how much the actual probability of a particular co-occurrence of two words together p(x, y) differs from what we would expect it to be on the basis of the probabilities of the individual events and the assumption of independence p(x)p(y).

$$\mathrm{i}(\mathrm{x}, \mathrm{y}) = \ln \frac{p(x,y)}{p(x)p(y)}$$

The perk of the Normalized PMI is that it has an upper and lower bound. Therefore, it is easier to quantify the similarity between the words.

$$\mathrm{i_n}(\mathrm{x}, \mathrm{y}) = (\ln \frac{p(x,y)}{p(x)p(y)}) / -\ln(p(x,y))$$

- NPMI = 1 when two words only occur together.
- NPMI = -1 when two words occur separately but not together.

Palmetto

In order to compute the NPMI, we used Palmetto[7].Palmetto is package java used to compute topic coherence measures between two words such as NPMI. Palmetto uses Wikipedia as its input corpus to compute the co-occurrence between the words for the NPMI.

Input: The file containing the top terms with one topic per line. In every line the top words of a certain topic are listed, separated by a single space.
Output: The jar will simply print out the topic's coherences between all the pair of words :

{word1, word2, NPMI-score}

2.5 MIIC

To generate the causal graph, we use an online service which aims at reconstructing a broad range of causal, non-causal or mixed networks from an observational data based of multivariate information statistics. The objective is to disentangle direct from indirect effects amongst correlated variables, including cause-effect relationships and the effect of unobserved latent causes. This service uses a Multivariate Information based Inductive Causation algorithm (MIIC[8]).

3

General View

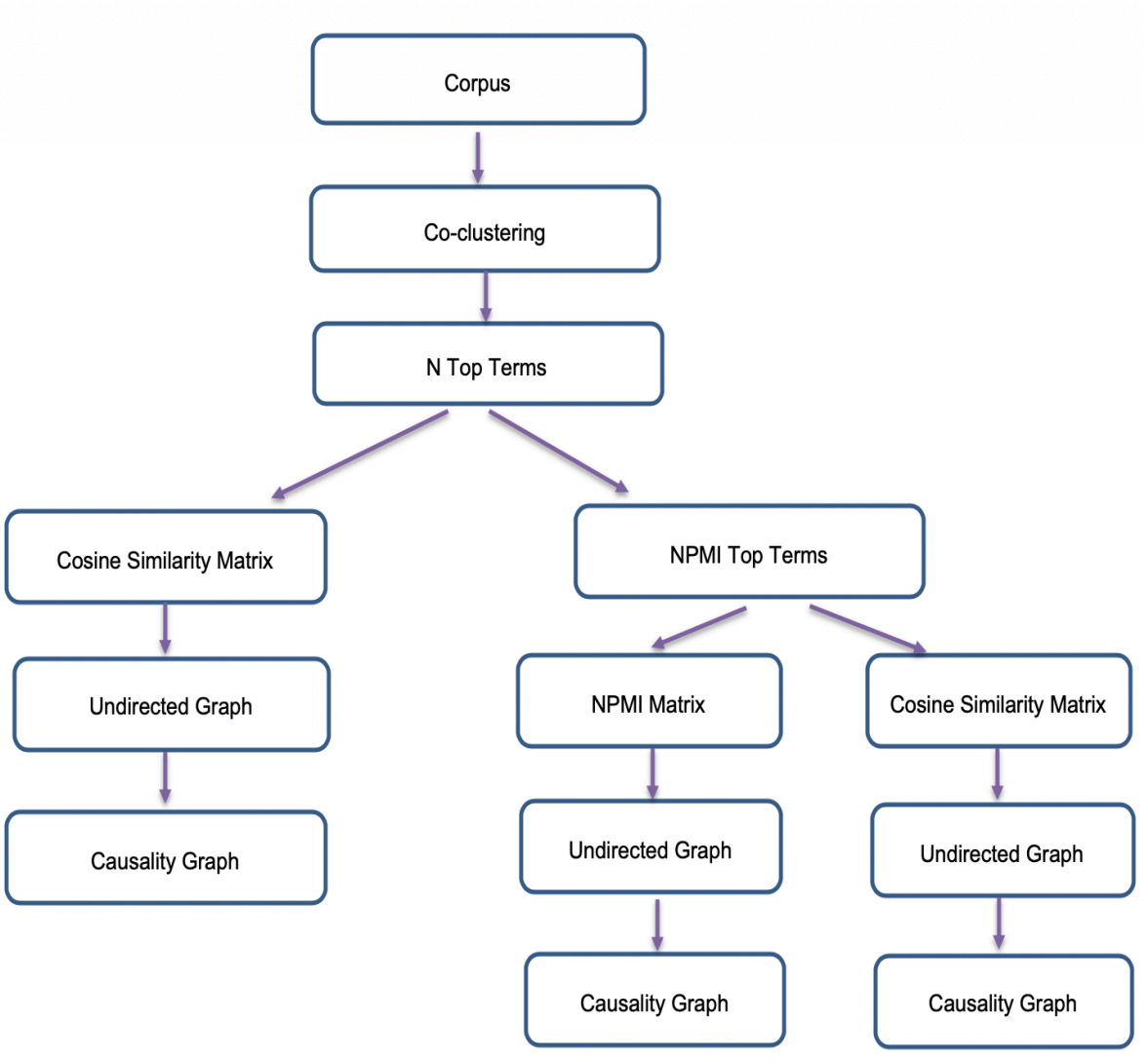

Figure 2: Schema of General View

4

General algorithm

4.1 Co-clustering applied to the corpus

As explained above we used the CoClustMod approach for the co-clustering of the adjacency matrix. The adjacency matrix gathers the abstracts of the 5 different diseases unsorted. Thus, after launching the co-clustering algorithm on all the abstracts, we obtain several sub-matrices, each one contains a specific topic about a disease.

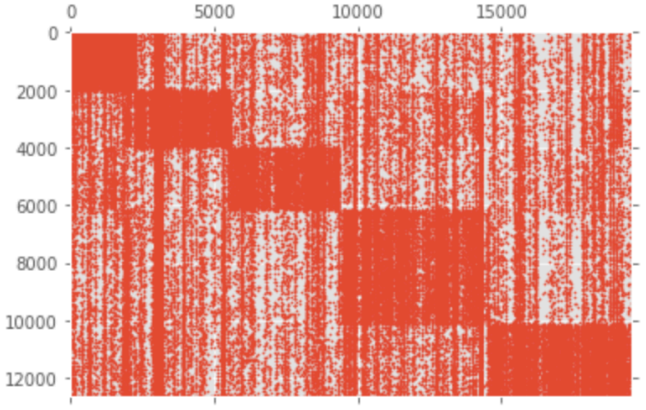

Figure 3: Co-clust on corpus of 5 diseases

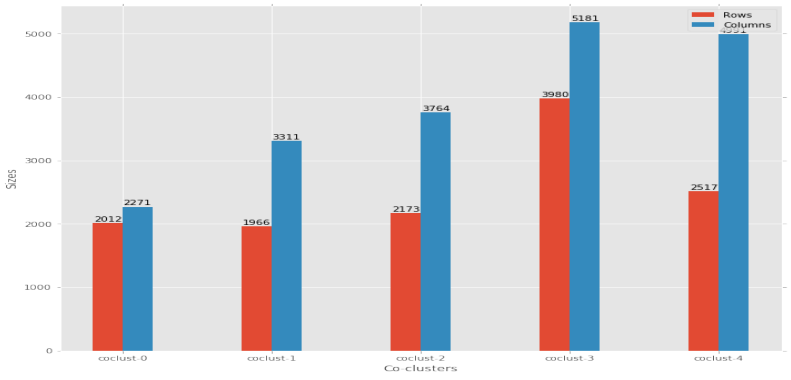

Figure 4: Number of rows and columns per cluster

4.2 Cosine graph with frequency

A first approach of obtaining the most significant terms of each disease, is to take the most frequently used ones in each topic of the corpus. We suppose that the more a term is used within the abstracts of a disease corpus, the more it is relevant.
Therefore, we seek the most important terms according to their frequencies in the set of abstracts by disease. Thanks to the co-clustering algorithm the operation is relatively simple. We order the frequencies of the terms within a certain cluster. We pick the n first terms, then we obtain the n-top terms of a cluster.

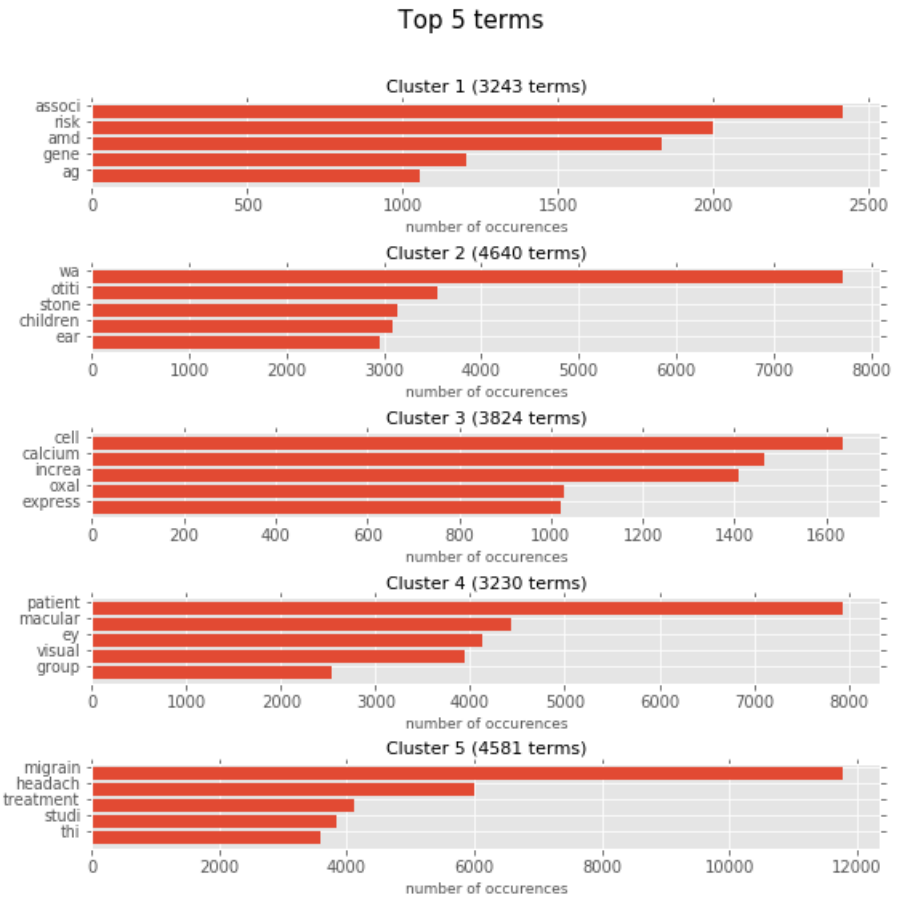

Figure 5: Top 5 terms according to frequency in corpus

Afterwards, in order to establish some links between the top terms of a certain cluster we compute a cosine similarity matrix. Therefore, for each element of the matrix we have a value included in the interval [-1,1] which defines the similarity between two words.
Using this cosine similarity matrix, we create a graph where the nodes represent the terms and the edges represent the link between the words. The diameter of the node is proportional to

the frequency of the term. Additionally, the color intensity of the edge indicates the degree of similarity between words. The darker the color is, the higher the similarity is between the two nodes.

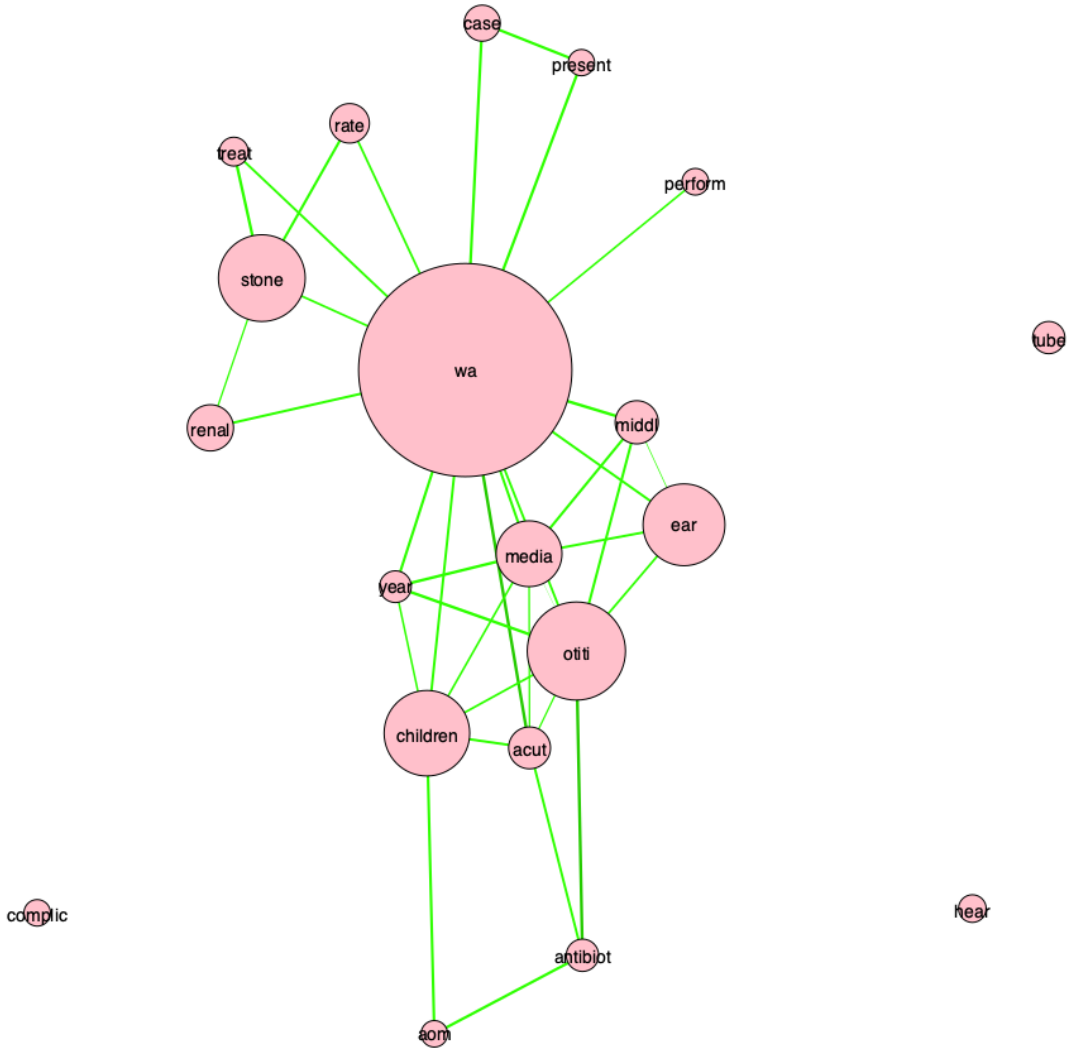

Figure 6: Example of a graph from cluster 2 - AMD

We notice some irrelevant terms such as "wa" are present in the n-top terms. Therefore, we will use another approach to determine significant words within a cluster.

4.3 Cosine graph with NPMI top terms

The graphs of the N-top terms according to their frequencies were satisfying. However, we noticed that some words are frequently used but are not relevant according to the disease. This means that a high frequency of a word does not imply its importance in the topic. Therefore, we tried another indicator to quantify the importance of a word in a corpus. The NPMI is used to measure the co-occurrence of two words together. So once Palmetto computes the NPMI for all the combinations of words. The algorithm attributes a certain score to each word. The score is the mean of the 5 highest NPMI score with other terms. In other words, this score represents the mean of the NPMI score of words which occur the most with the current one. Finally, the words are sorted according to their score.

Then we calculate the cosine similarity matrix to generate the graph.

In order to have 20 significant word, we use the Palmetto algorithm on 30 top words and pick only the 20 first words. This will guarantee that the 10 least significant words will not be taken into consideration in the graph.
To conclude, in this approach a word is significant if it occurs a lot with other words.

0	-0,26514	[wa, macular]
1	-0,04066	[wa, eye]
2	-0,01769	[wa, visual]
3	-0,32570	[wa, amd]
4	-0,30989	[wa, retinal]
5	-0,27498	[wa, acuity]
6	-0,03622	[wa, associate]
7	-0,30726	[wa, degeneration]
8	-0,02635	[wa, group]
9	-0,15850	[wa, neovascular]
10	-0,06746	[wa, significant]
11	-0,06592	[wa, month]
12	-0,30795	[wa, diabetic]
13	-0,00970	[wa, no]
14	-0,07391	[wa, improve]
15	-0,43949	[wa, risk]
16	-0,14115	[wa, choroiditis]
17	-0,29578	[wa, edema]
18	-0,45692	[wa, measure]

Figure 7: NPMI between a pair of words

```
0 macular -->  ['degeneration', 'neovascular', 'edema', 'retinal', 'diabetic'] [0.80604 0.62993 0.58788 0.53307 0.485
12]
1 degeneration -->  ['macular', 'retinal', 'amd', 'diabetic', 'eye'] [0.80604 0.55617 0.4049  0.37944 0.19901]
2 retinal -->  ['degeneration', 'macular', 'edema', 'acuity', 'diabetic'] [0.55617 0.53307 0.42495 0.39092 0.38384]
3 edema -->  ['macular', 'retinal', 'diabetic', 'injection', 'associate'] [0.58788 0.42495 0.36009 0.21387 0.15407]
4 diabetic -->  ['macular', 'retinal', 'degeneration', 'edema', 'injection'] [0.48512 0.38384 0.37944 0.36009 0.2295
6]
5 acuity -->  ['visual', 'retinal', 'eye', 'measure', 'improve'] [0.54285 0.39092 0.29803 0.20897 0.18933]
6 visual -->  ['acuity', 'retinal', 'macular', 'eye', 'degeneration'] [0.54285 0.29586 0.23859 0.17327 0.13277]
7 amd -->  ['macular', 'degeneration', 'risk', 'factor', 'development'] [0.47831 0.4049  0.07705 0.05191 0.04979]
```

Figure 8: The 5 highest NPMI scores for each word

	0	1
0	macular	0.608408
1	degeneration	0.469112
2	retinal	0.457790
3	diabetic	0.367610
4	edema	0.348172
5	acuity	0.326020
6	visual	0.276668
7	eye	0.250324
8	injection	0.233062
9	risk	0.227270
10	amd	0.212392

Figure 9: The top terms sorted according to their NPMI score

Top terms with frequency

”wa”, ”macular”, ”eye”, ”visual”, ”amd”, ”retinal”, ”acuity”, ”associate”, ”degeneration”, ”group”, ”neovascular”, ”significant”, ”month”, ”diabetic”, ”no”, ”improve”, ”risk”, ”choroiditis”, ”edema”, ”measure”, ”intravitreal”, ”result”, ”thick”, ”change”, ”injection”, ”case”, ”evaluate”, ”development”, ”show”, ”factor”

Top terms with NPMI scores

“macular”, “degeneration”, “retinal”, “edema”, “diabetic”, “acuity”, “visual”, “amd”, “injection”, “eye”, “neovascular”, “risk”, “factor”, “intravitreal”, “significant”, “measure”, “associate”, “evaluate”, “improve”, “result”, “change”, “development”, “case”, “thick”, “show”, “choroiditis”, “group”, “month”, “no”,”wa”

In the example above we can see that, words such as “wa” moved from the first rank to the last one. We could not find what this word’s meaning in the literature. Therefore, we can suppose that the word is not significant in the topic. Moreover, we can see in the example above that it has a low NPMI score with most of the other terms.

We also noticed that the word “eye” moved a few ranks back, while “retinal” moved few ranks forward. That is because the word “retinal” is more specific to the Aged Macular Disease in comparison to the word “eye”.

4.4 NPMI GRAPH

In this part instead of using the cosine similarity matrix, we use the NPMI matrix to generate the graph. The NPMI matrix contain the NPMI-score for each pair of words. So, the edges in the graph will appear according to the NPMI score between two words, if it is higher than the threshold an edge will appear between the two words. Therefore, an edge will represent a high co-occurrence between two words.

Unnamed: 0	otitis	infection	pneumonia	bacterial	acute	chronic	antibiotics	effusion	recurrent	complication	ear	influenza
otitis	0.00000	0.48701	-0.28542	0.44146	0.22219	0.47456	0.39581	-0.06334	0.26187	-0.20776	-0.22570	0.21981
infection	0.48701	0.00000	0.02468	0.01163	0.18929	0.14070	0.16808	-0.22721	0.15994	-0.37163	-0.38957	0.28045
pneumonia	-0.28542	0.02468	0.00000	0.02054	-0.04180	0.01464	-0.37793	-0.26232	-0.05683	-0.40675	-0.42468	-0.00602
bacterial	0.44146	0.01163	0.02054	0.00000	-0.00231	0.02749	-0.40251	-0.28690	-0.07883	-0.02664	0.10234	-0.01591
acute	0.22219	0.18929	-0.04180	-0.00231	0.00000	-0.02520	-0.41891	-0.30330	0.02159	-0.44773	-0.06308	-0.02461
chronic	0.47456	0.14070	0.01464	0.02749	-0.02520	0.00000	0.24369	-0.20617	0.04605	-0.35060	-0.36854	0.09826
antibiotics	0.39581	0.16808	-0.37793	-0.40251	-0.41891	0.24369	0.00000	-0.15585	-0.35872	-0.30027	-0.31821	-0.40248
effusion	-0.06334	-0.22721	-0.26232	-0.28690	-0.30330	-0.20617	-0.15585	0.00000	-0.24311	-0.18467	-0.20260	-0.28687
recurrent	0.26187	0.15994	-0.05683	-0.07883	0.02159	0.04605	-0.35872	-0.24311	0.00000	-0.38753	-0.40547	-0.01582
complication	-0.20776	-0.37163	-0.40675	-0.02664	-0.44773	-0.35060	-0.30027	-0.18467	-0.38753	0.00000	-0.34703	-0.01775
ear	-0.22570	-0.38957	-0.42468	0.10234	-0.06308	-0.36854	-0.31821	-0.20260	-0.40547	-0.34703	0.00000	-0.03993
influenza	0.21981	0.28045	-0.00602	-0.01591	-0.02461	0.09826	-0.40248	-0.28687	-0.01582	-0.01775	-0.03993	0.00000
pneumococcal	0.55789	0.27857	-0.31280	0.15831	0.12738	-0.25665	-0.20632	-0.09071	-0.29358	-0.23514	-0.25308	-0.33735
cause	-0.41047	-0.07288	0.01836	-0.04083	-0.01163	-0.05593	-0.06762	-0.38738	-0.10440	-0.06063	-0.04028	-0.03759
media	0.43285	0.14717	-0.02271	-0.00034	-0.02057	0.50645	0.23682	-0.20632	0.02685	-0.35075	-0.36869	0.00012
hear	-0.19659	-0.36046	-0.39558	-0.42016	-0.43656	-0.33943	-0.28910	-0.17349	-0.37636	-0.31792	-0.33586	-0.01539
isolate	0.44025	0.30489	-0.04899	-0.01170	-0.00721	0.36733	0.45937	-0.22907	0.08327	-0.37350	-0.39143	-0.00455
middle	-0.35600	0.01146	0.01973	-0.01593	-0.01417	0.04811	0.00486	-0.33290	-0.04112	-0.04603	-0.02505	-0.00158
tube	0.19867	0.09447	-0.04627	-0.04009	-0.03995	0.20786	0.16889	-0.32099	0.06649	-0.46541	-0.09341	0.03025
vaccin	-0.25862	0.03046	-0.45761	-0.09737	0.01517	0.08821	0.19103	-0.23553	0.04461	-0.37995	-0.39789	-0.01468

Figure 10: Example of NPMI matrix

4.5 Directed Graph

After obtaining the undirected graphs, we wanted to enhance the information in the graph by adding causality links between the words. As it was mentioned before, we used an online service to generate the causal graph with the MIIC algorithm. The algorithm takes as an input a file with the terms with their embeddings. The file contains a model with the significant words as variables and their dimension as the rows of the model. The vectors should have large dimensions to obtain a directed graph.

For the top terms we used the top terms we got with the NPMI because they are more accurate. Then for the embeddings, after few tests, we kept the ones with the most medical background, such as the embeddings from PMC and the embeddings from Wikipedia. So, the final embeddings were 800-dimensional vectors. We concatenated PMC and Wikipedia embeddings learned with Word2Vec, plus Wikipedia and Gigaword learned with FastText skipgram and finally embeddings learned on Wikipedia with Word2Vec continuous skipgram.

5

Results of algorithm and interpretation

5.1 Undirected graph

We used several methods to create the graphs to express the relationship between the terms of different diseases. Indeed, for each disease we created graphs according to 3 different routes as explained above. In this section, we are going to discuss only two diseases which are: AMD and Otitis. We will compare the different undirected graphs of the algorithms. For each route and for each disease, several undirected graphs were created according to a base of embeddings:

- Embeddings from Glove
- Embeddings from Gensim
- Embeddings from FastText
- Embeddings from Word2Vec using PMC texts
- Embeddings from the concatenation of all the embeddings mentioned before

AMD

Graphs created from FastText embeddings and Gensim embeddings were able to show the links between ”diabetes”, ”Retina” and ”Choroid”. It is known in the medical field that diabetes can cause diabetic retinopathy. It can cause eye pain and retinal problems. Additionally, according to some studies, individuals with diabetic retinopathy have an alteration in the thickness of the choroid. Unfortunately, the PMC Embeddings were not as efficient to show the link between “diabetes” and the words related to the disease.

Diabetic macular edema results from an inflammatory process at the back of the eye and happens when the small blood vessels in the retina are damaged. The flow of liquid between the different cell layers causes swelling in the center of the retina, also called the macula, the area of maximum visual acuity of the eye. This results a blurred vision and loss of acuity which has an impact on daily life since people with this disorder have difficulty reading and driving. We also find some links between «acuity”, ”macular” and ”diabetic” in the graphs below.

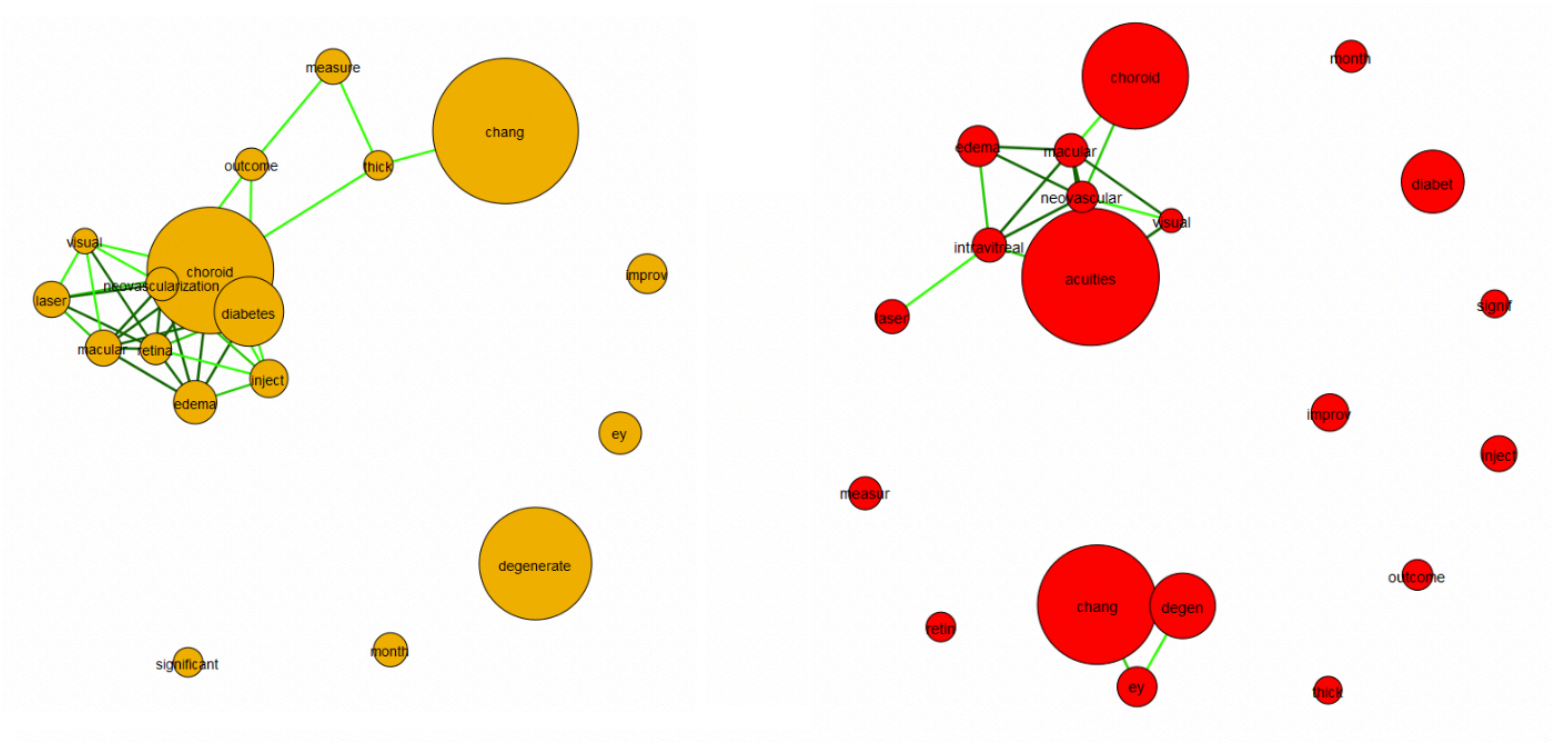

Figure 11: Graphs from Gensim(Left) and from Word2Vec training on PMC(Right) using frequency

The results of the n top terms according to the NPMI score are much different. However, the graph of the PMC embeddings gave better results. Indeed, the graph showed the Link between AMD and "diabetes". However, the concatenation of all the embeddings gave better results. Indeed, we can see in the graph that the algorithm was able to separate the words into two clusters. the first one is about all the significant words related to the AMD disease. The second cluster is about common words used in the scientific field. These words are not very significant in our case.

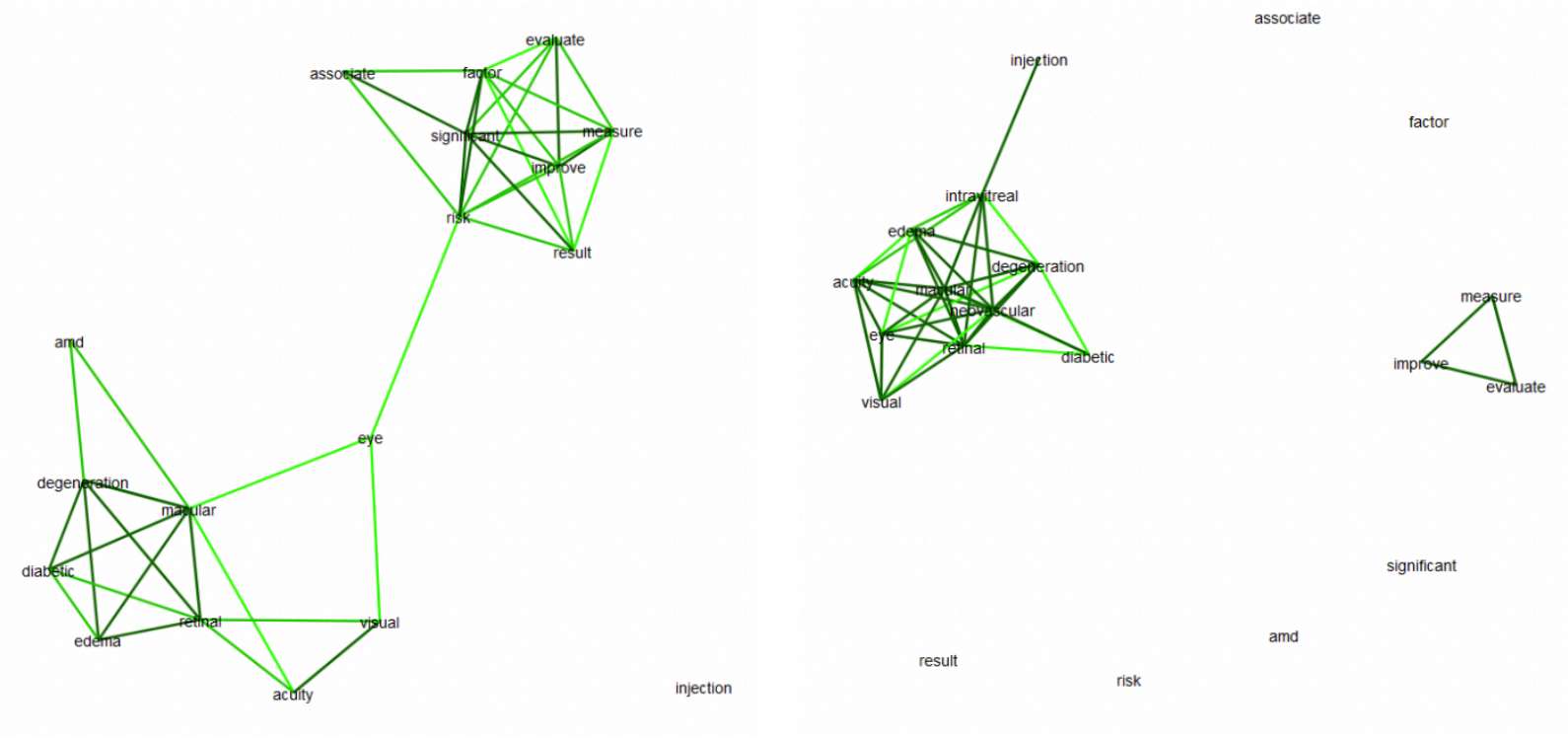

Figure 12: Graphs from mergin of embeddings(Left) and from Word2Vec training on PMC(Right) using NPMI

For the graph below we used the NPMI matrix to display the links between the words. The words are linked if they have a high common NPMI score. We can notice some interesting links such as degeneration-diabetic-injection-improve. Adults with type 2 diabetes use medicine injections to improve their blood sugar reduce the risk of major cardiovascular events. We can also see that the NPMI matrix was able de detect a link between AMD and diabetes even though the link is not strong enough.

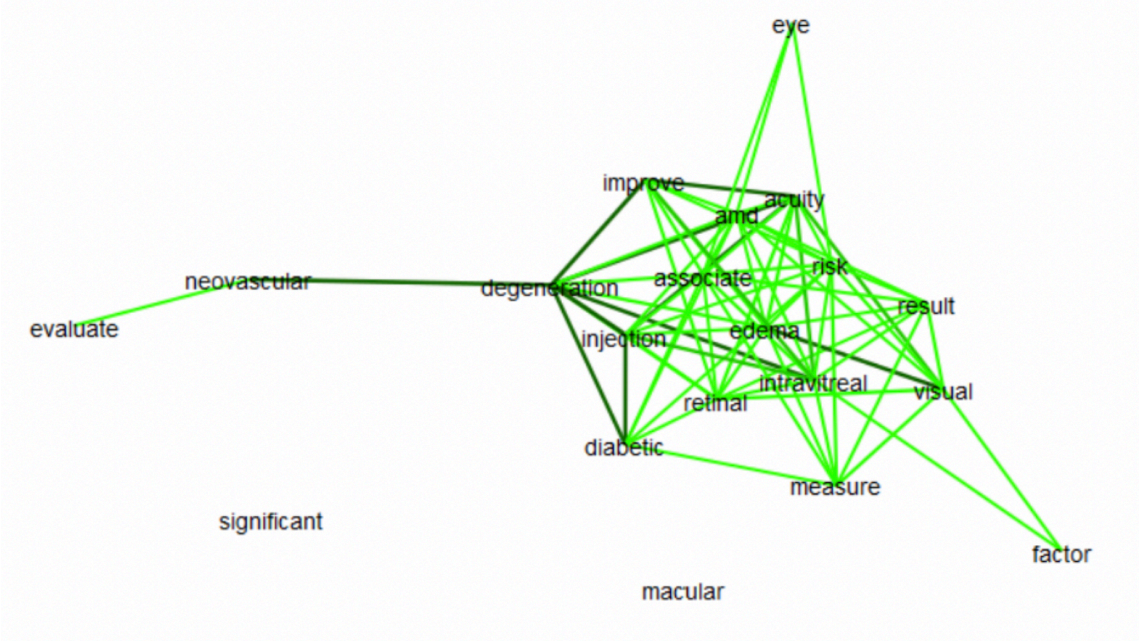

Figure 13: Graph from merging of embeddings using NPMI matrix

Otitis

The graph with the most frequent words gave some satisfying results, especially with the PMC embeddings. The graph shows a link “middle”, “ear”, “tube”. This link reflects the ”middle ear disease” which causes hearing loss, dizziness, or painful middle ear inflammation. There is also a disease called ”otitis media with effusion” that mainly affects children. We can find these links on the graph generated by the Gensim and PMC embeddings, which are embeddings with a medical background.

Additionally, allergy to pollen can lead to complications such as sinusitis or an infection of the middle ear commonly known as Otitis. This information can be found in the generated by Gensim, FasText and Glove embeddings but not from PMC. The lack of the information is due to a lemmatization problem. Indeed, we can see that in the graph the word “allergy” is replaced by “allerg”. Thus, the embeddings of the words are not similar which leads to a different similarity with the other words.

Again, the concatenation of the four embeddings gave the best results.

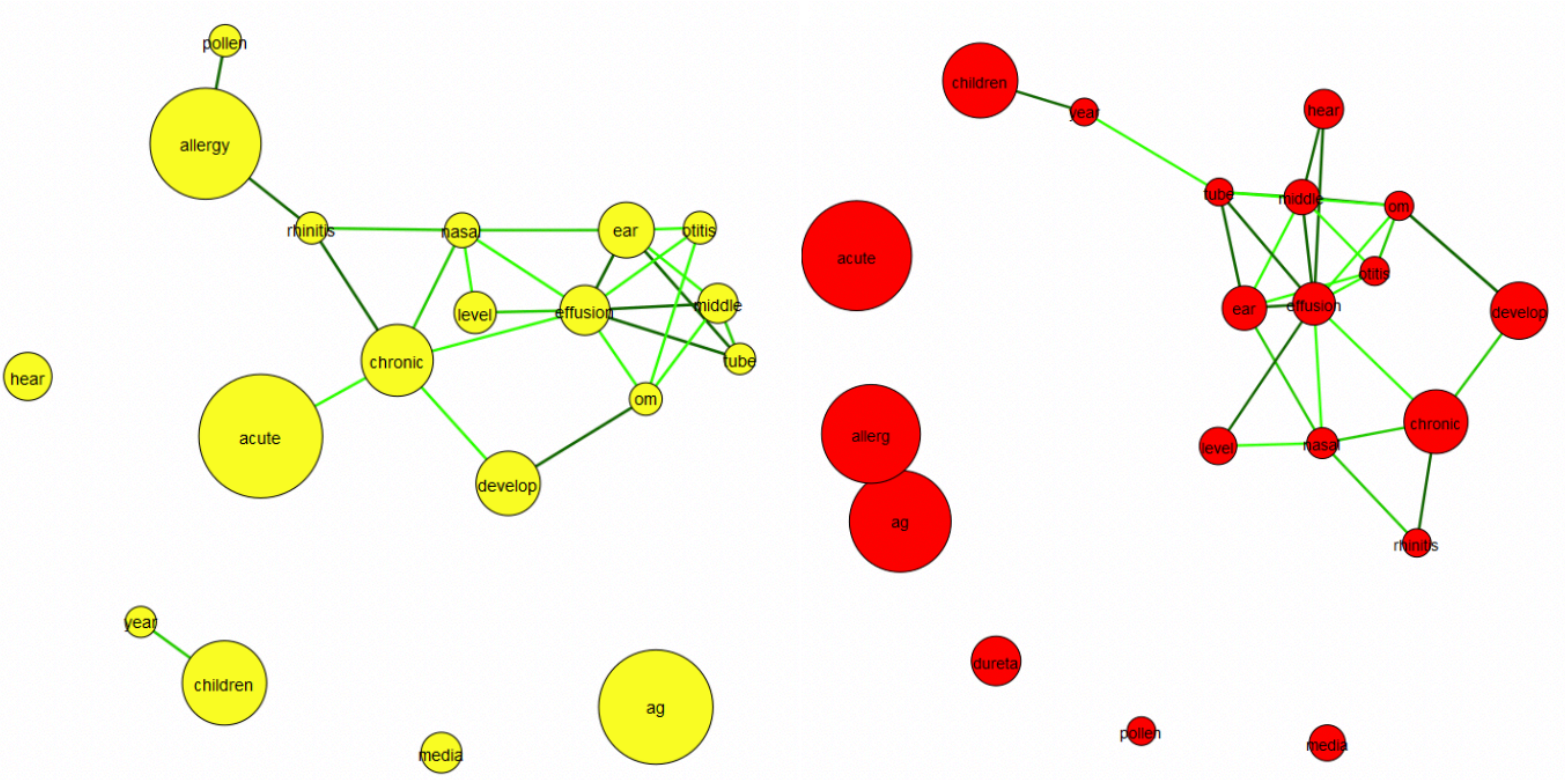

Figure 14: Graphs from margin of embeddings(Left) and from Word2Vec training on PMC(Right) using frequency

The words we obtained using the NPMI are more relevant. Therefore, the graph is more informative. We notice a link between these following words: pneumococcal - bacterial - infection - vaccine - chronic.

In fact, pneumococcal infections most often affect fragile people (people with chronic diseases, young children, seniors, etc.) but can be avoided thanks to vaccination. They are caused by a bacterium called Streptococcus pneumoniae. They can cause diseases like otitis.

We can also find the link between “middle-ear-tube” reflecting the ”middle ear disease”.

Unifying the four embeddings, allows us to extract some relevant information from the graph.

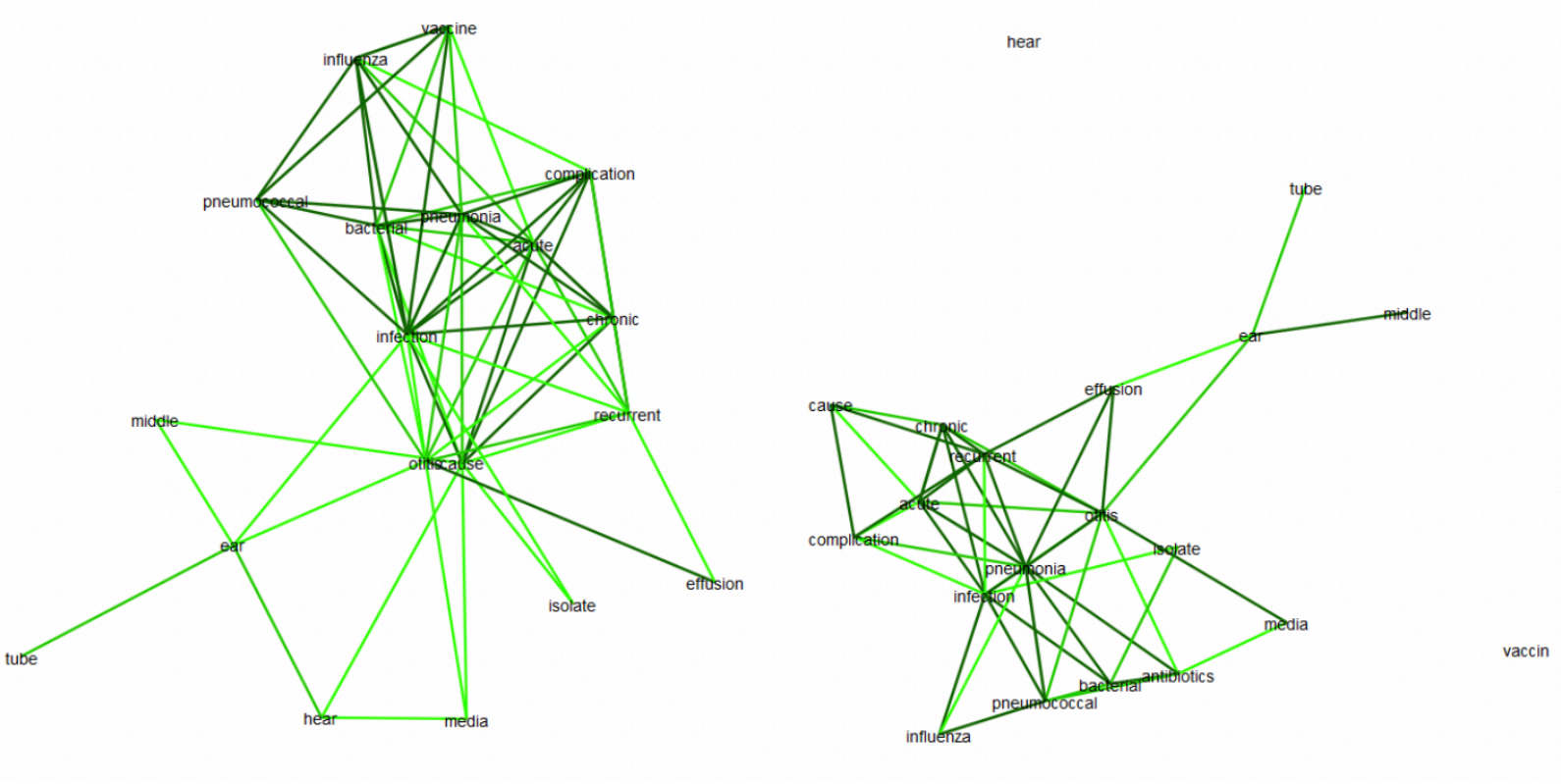

Figure 15: Graphs from margin of embeddings(Left) and from Word2Vec training on PMC(Right) using NPMI

From the graph below we can see that the NPMI matrix provides the same information as the cosine similarity matrix but with less precision. However, we distinguish few strong links such as otitis-pneumococcal-bacterial-infection-antibiotics. These terms are clearly highly related.

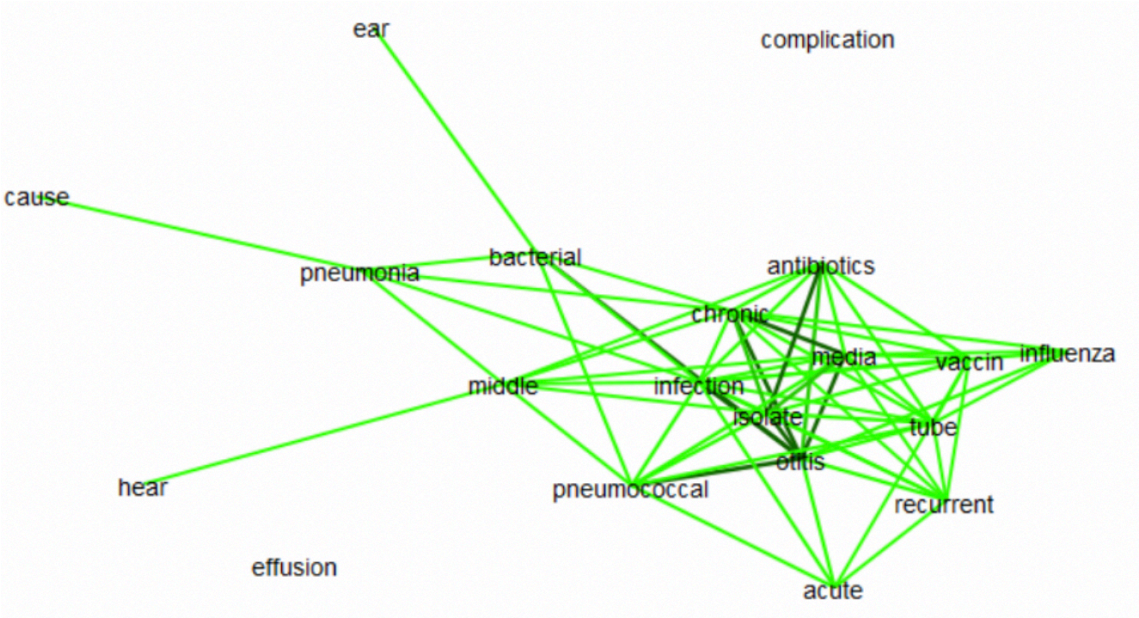

Figure 16: Graph from merging of embeddings using NPMI matrix

To conclude this section, we can say that the source of the embeddings plays a major role in linking the significant words. Indeed, the embeddings generated by Gensim, PMC and FastText were trained from Wikipedia which contains a lot of scientific articles unlike the embeddings generated by Glove trained from GigaWord which is mainly composed by journals and newspapers.
Embeddings generated by Word2Vec are trained from the PubMed Central corpus. This corpus includes over 30 million citations for biomedical literature from MEDLINE, life science journals, and online books.
Thus, it provides more specific information compared to the other embeddings. However, the trained corpus is exclusively biomedical. In order to get as much information as possible, we merged the PMC embeddings with some embeddings trained on common corpora.
So, by merging the different sources of embeddings, we recover the best of each embedding and obtain better results.
As a conclusion, we can say that the NPMI gives the best result for extracting the most relevant terms. Additionally, the cosine similarity and the NPMI were efficient to determine the links between the top words. Therefore, the graph with the most satisfying results is the cosine graph with the NPMI top terms.

5.2 Directed graph

In this project directed graphs are used to add the information of the causality between the words. After some tests we noticed that in order to have causality links between the words we had to fulfil two conditions. The first one is that the dimension of the vectors should be large. The second condition is that, the embeddings should have enough medical background. Therefore, we used the embeddings with the most medical background such as the embeddings from PMC and the embeddings from Wikipedia. So, the final embeddings were 800-dimensional vectors. We concatenated PMC and Wikipedia embeddings learned with Word2Vec, plus Wikipedia and Gigaword learned with FastText skipgram and finally embeddings learned on Wikipedia with Word2Vec continuous skipgram.

We obtained the following graphs:

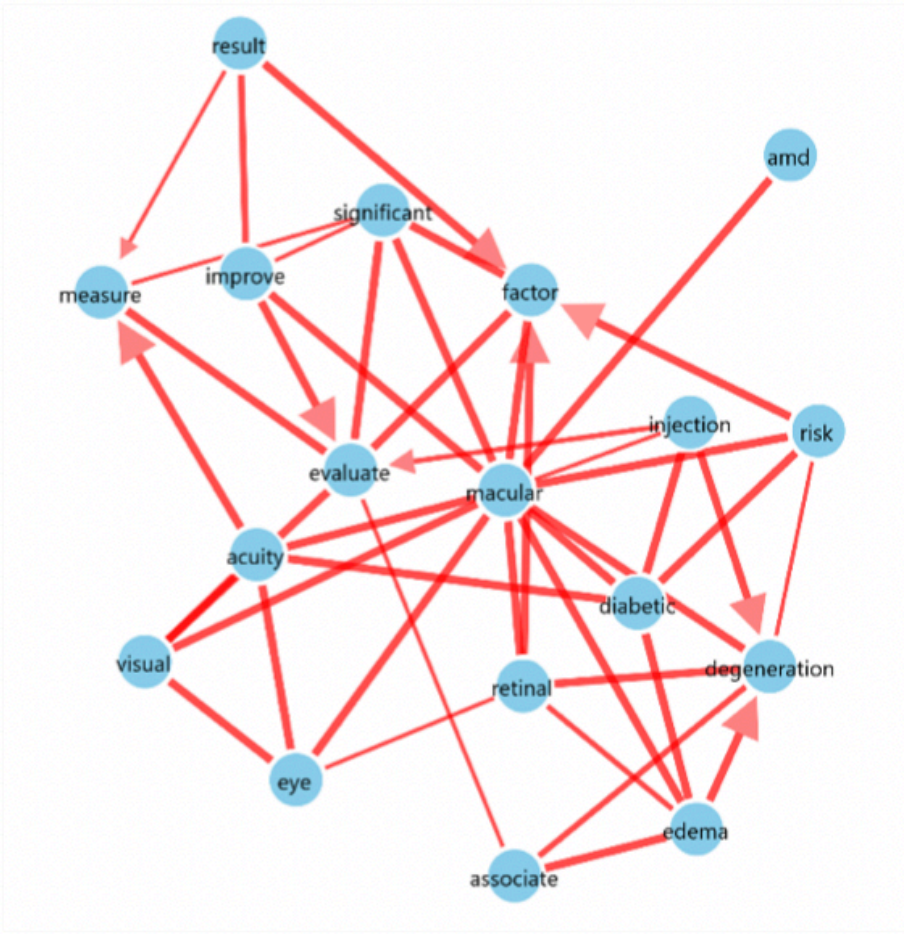

Figure 17: Directed graph on AMD

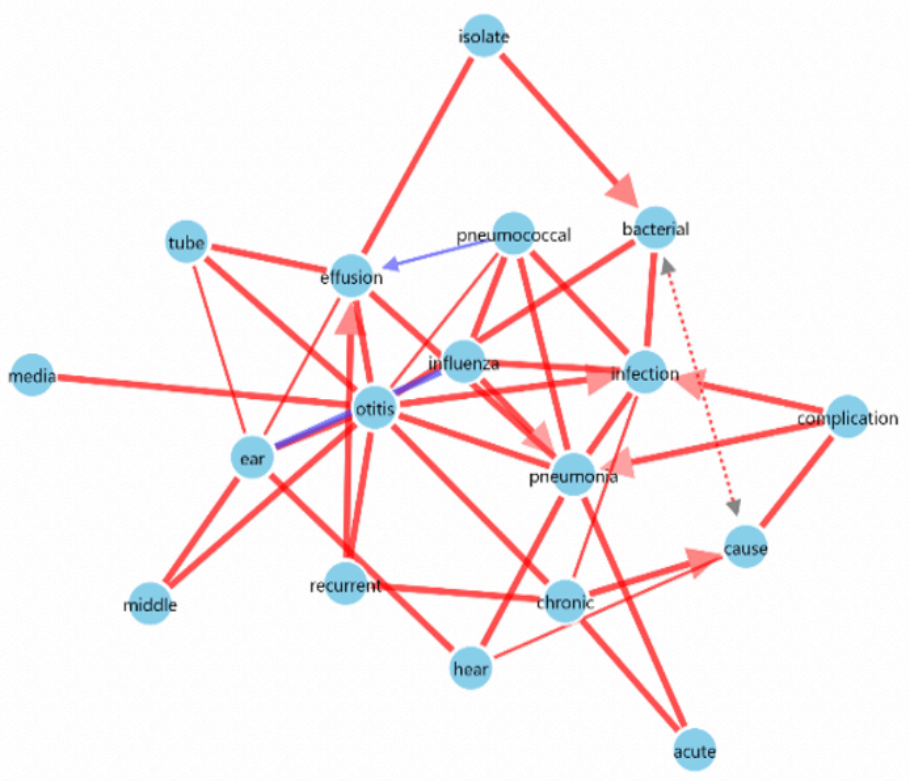

Figure 18: Directed graph on Otitis

Unfortunately, in the AMD graph we do not have many causal links between the medical words. For example, there is an arrow from "edema" to "degeneration". Edema is defined medically as the swelling from fluid accumulation in the body tissues, it is usually due to a certain degeneration of a certain organ in this case it's the eye. Therefore, what we can say about the two words is that if the word "edema" is in a text, there is high probability that it will be followed by the word "degeneration". We can also notice some arrows from "risk" and "result" to "factor".

The Otitis graph shows more causality links between the medical words.
For example:

- "infection" -> "otitis": Otitis is an inflammation of the ear. An inflammation is the biological response of body tissues to harmful stimuli such as an infection. So here we can also detect a medical causality between the two words which is that an otitis can be caused by an infection.

- "influenza" -> "pneumonia": Pneumonia is an acute disease that is marked by inflammation of lung tissue. Bacterial Pneumonia can be an influenza complication according

to some articles on PubMed. So, influenza can cause pneumonia.

- "Pneumococcal" -> "effusion": Pneumococcal is related to the bacteria pneumococcus which is a bacteria that causes many infections such as ear infections. Effusion is an outpouring of fluid which can be caused by an infection and in this case an otitis.
- "bacterial" ... "cause": The link between the two words is different than the others. That is because the words are the cause of a third word that the algorithm could not find.

5.3 Evaluation of the embeddings

Since we took the decision to upload the embeddings from the internet, we wanted to evaluate them in order to test their accuracy. We used the function "evaluate_word_pairs()" which computes the correlation with human opinion on word using Keyedvectors. This function takes a file as parameter. In the file we find pairs of words with a score which represents the similarity between the words set by a human. Then this score is compared to the one computed with the embeddings. The function measures the correlation between both scores.

Here, we have decided to use Spearman correlation because it is more accurate. We used three different files for these evaluations. We obtained the following results:

	Glove	**Word2Vec**	**FastText**	**PMC**
Wordsim353	0.555504	0.673403	0.672899	0.529629
Wordsim_relatedness_goldstandard	0.535715	0.61602	0.61602	0.540422
Worldsim_similarity_goldstandard	0.589795	0.708539	0.696419	0.540508

The correlation scores are high for the Word2Vec and The FastText embeddings. However, the score is slightly low than the others for the Glove Embeddings. The difference between the evaluation of the embeddings is due to the volume of the corpus they were learned from.

For PMC, the score is relatively low. That is because the files used for evaluation do not have medical pairs of words. Therefore, we used 3 files with medical terms only. These files allow us to properly evaluate the embeddings for the purpose of this project.

	Glove	**Word2Vec**	**FastText**	**PMC**
MayoSRS	0.389869	0.223434	0.217196	0.395553
UMNSRS_relatedness	0.478409	0.306229	0.348013	0.501167
UMNSRS_similarity	0.481655	0.341951	0.34444	0.560862

Here we can see that PMC provides better results for these evaluations. Surprisingly, Glove gives better results than FastText and Word2Vec which means that it also has a medical background. However, according to the results of the graph we can say that its medical background is not large enough to provide specific information.

6

Conclusion

The purpose of this project is to facilitate the extraction of relevant information from large corpus of biomedical documents. The kind of information we seek is the causalities between the most relevant terms of a certain disease. We worked on the five following diseases: Hay fever - Kidney calculi - Age-related Macular Degeneration - Migraine - Otitis.

We applied a co-clustering algorithm on the adjacency matrix of PubMed5. As a result, we got a matrix with 5 sub-matrices, each one corresponds to a specific disease.

In order to retrieve the most relevant terms from a corpus of documents of a disease. We used two different approaches. The first one is the word's frequency and the second one is the NPMI score. The first approach gave some satisfying results. However, we noticed some frequent words which are not significant according to the disease. The first results led us to the conclusion that a high frequency does not imply a word's importance in the topic. The NPMI score on the other hand gave better results in terms of extracting the most significant words.

Once we get the most significant words, we try to display them in the most informative way which is the graph. Indeed, a non-directed graph allows us to represent the most significant words and the links between them. The link between two words is determined by their similarity, and for this we chose the cosine similarity. To represent the words, we used 5 different embeddings, each embedding was learned from a different corpus using different algorithm. The embeddings with the best results are the one with the medical background. The last step of this project was the causal graph. The causal graph is used to represent the causal link between the significant words of the disease. After some experiments, we noticed that in order to have causality links between the words we had to fulfil two conditions. The first one is that the dimension of the vectors should be large. The second condition is that, the embeddings should have an important medical background. Therefore, we used the embeddings

with the most medical background such as the embeddings from PMC and the embeddings from Wikipedia. Thus, we merged all the embeddings except the embeddings learned from Gigaword which is a corpus of newspaper articles with no medical background. Using the merged embeddings, we obtain a causal graph with many causal links between the words. These links translate medical causality which can easily be interpreted by a professional.

In this project, we used available embeddings on the internet. These embeddings were 200 and 300-dimensional vectors. These dimensions may not be optimal. For future improvements, we can generate embeddings with the same algorithm and the same dataset for learning but with different dimensions. Indeed, the dimension of the embeddings will commensurate with their effectiveness in providing medical information. We can adjust the dimensions of the embeddings according to their evaluation score which is Spearman correlation.

At last, we used the package Palmetto to calculate the NPMI score between the words. Palmetto uses Wikipedia as an input dataset to compute the co-occurrence between the words. It would be interesting to see the difference if we use another tool which uses a medical dataset as an input.

To conclude the algorithm presented in this paper was able to extract some medical information from a corpus of biomedical of different diseases. However, the algorithm can also be used on a corpus with one specific disease. This will allow the user to extract information with different perception of the disease.

7

References

[**1**]: François Role, Stanislas Morbieu, Mohamed Nadif. CoClust: A Python Package for Co-clustering. 2018. ffhal-01804528f.

[**2**]: Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. In Proceedings of Workshop at ICLR, 2013.

[**3**]: Pennington, J., Socher, R., Manning, C. D. (2014, October). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).

[**4**]: Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., Joulin, A. (2017). Advances in pre-training distributed word representations. arXiv preprint arXiv:1712.09405.

[**5**]: Rahutomo, Faisal Kitasuka, Teruaki Aritsugi, Masayoshi. (2012). Semantic Cosine Similarity.

[**6**]: Bouma, G. (2009). Normalized (pointwise) mutual information in collocation extraction. Proceedings of GSCL, 31-40.

[**7**]: paper: Röder, M., Both, A., Hinneburg, A. (2015, February). Exploring the space of topic coherence measures. In Proceedings of the eighth ACM international conference on Web search and data mining (pp. 399-408).

[**8**]: Sella, N., Verny, L., Uguzzoni, G., Affeldt, S., Isambert, H. (2018). MIIC online: a web server to reconstruct causal or non-causal networks from non-perturbative data. Bioinformatics, 34(13), 2311-2313.