Utilizing AutoPhrase on Computer Science papers over time

Cameron Brody
crbrody@ucsd.edu

Jason Lin
jylin@ucsd.edu

James Yu
jjy002@ucsd.edu

Abstract

Phrase mining is a useful tool to extract quality phrases from large text corpora. Previous work on this topic, such as AutoPhrase, demonstrates its effectiveness against baseline methods by using precision-recall as a metric. Our goal is to extend this work by analyzing how AutoPhrase phrases change over time, as well as how phrases are connected with each other by using network visualizations. This will be done through exploratory data analysis, along with a classification model utilizing individual phrases to predict a specific year range.

1 Introduction

Phrase mining is the process of utilizing automated programs for extracting important and high-quality phrases from bodies of text. These phrases can be used in a variety of ways, from extracting major ideas from customer reviews or key points from a scientific paper. However, phrase mining has historically been done with complicated linguistic analyzers trained on specific data, meaning that it is difficult to expand to a larger scope without significant additional human effort. As a way to mine phrases in an expandable way, in any language or domain, AutoPhrase was created. With AutoPhrase, it is possible to input any text corpora without the need for human labels, allowing for much faster extraction of phrases in a variety of documents.

With that in mind, we utilized AutoPhrase to extract the phrases from a database of 3,079,007 computer science research papers aggregated from 1950 to 2017. With this, we can trace the evolution of key ideas through the history of computer science, as well as find which ideas were most common in what years and how they connect with each other. Additionally, we used the extracted phrases as data to construct a classification model for finding what year a paper belongs to based on its key phrases as a way of showing how strong the connections are between ideas and time.

2 Methods

2.1 Data gathering and processing for DBLP v10 dataset

Our initial goal was to gather data on Computer Science papers over time, looking at titles, abstracts, and paper contents. However, we realized that gathering and working with full paper text would result in much larger and messier data, while likely not benefiting the results of AutoPhrase and our model. As a result, we chose to focus on the DBLP Computer Science Bibliography dataset. We chose this dataset as it contains a large amount of papers (3 million+) with information on each paper's title, abstract, and publication year. These attributes are all that is needed for the purposes of our analysis. There are 13 versions of the dataset, but ultimately we chose to focus on the v10 dataset.

Our initial data processing was done on both the DBLP v10 and v13 datasets. The v13 dataset is the latest version of the DBLP dataset from AMiner, released in May of 2021 with over 5 million papers. It contains all of the information previously specified, but it also includes keywords for each paper. We thought this would be beneficial as it allows for a point of comparison against the phrases we

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

would extract in the future by utilizing AutoPhrase. However, the v13 dataset had many issues with formatting that caused issues when trying to process it. The entire dataset is contained in a .json file that is too large to store in memory, so we had to process it line-by-line. However, the information for each paper is not contained on a single line–rather, it is spread out across multiple lines. This results in issues while processing each paper, as there are formatting issues that need to be resolved with many different cases.

The DBLP v10 dataset has fewer papers compared to v13 as it was released in 2017, but it still has information on 3 million+ papers. Additionally, it is much easier to work with as the information for each paper is stored in a single line. We created a function that processes the dataset and outputs the relevant information into .txt files in preparation for phrase mining. As we want to examine how phrases change over time, papers are grouped together based on their publication year. Ultimately, we decided to group years together in groups of 5, as we believe a single year may not be a significant marker of change in the Computer Science field overall. By using intervals of 5, we can obtain a clearer picture of the general trend of phrases and the change over a longer period of time.

All years are grouped in groups of 5 years, except for the last years in the dataset (2015-2017) and the beginning years of the dataset (1950-1959). We decided to group the earlier years together in a larger group as there are not as many papers in the earlier years.

When processing the papers, we realized that there were quite a few papers with empty abstracts or invalid years. We chose to exclude any papers with empty abstracts and invalid years from the output .txt files. We specified invalid years as anything prior to 1950 and anything after 2017. In total, there were 530,394 papers with empty abstracts, and 82 papers with invalid years.

2.2 Exploratory data analysis for DBLP v10

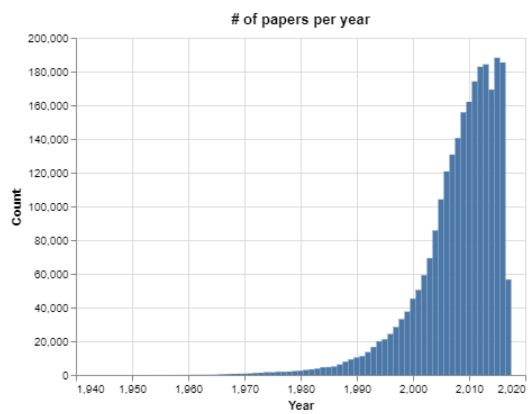

Figure 1: Document count for DBLP v10

Figure 1 only includes information on papers that were included in the output files for phrase mining. So the papers with empty abstracts or invalid years were not included. DBLP v10 contains 3,079,007 papers, but from our data processing steps, we filtered out 530,394 of the papers for having empty abstracts, and 82 of the papers for having irrelevant years (anything before 1950). Thus, this graph shows the distribution of the remaining 2,548,531 papers.

The number of papers has increased exponentially in recent years as the Computer Science field has grown, both in popularity and complexity. There are many more sub-fields to explore and develop. Overall the distribution has been strictly increasing over time, with the exception of a small dip in 2014, and the right-most bar in the graph representing 2017. It is smaller than would be expected as the DBLP v10 dataset was published before the year was over, meaning it does not actually contain all of the papers published in that year.

2.3 Running AutoPhrase: Phrase mining and Phrasal segmentation

AutoPhrase has two functions that can be run. The first, phrase mining, is the process described in the Introduction. AutoPhrase only requires a single .txt file and it will output another .txt file containing the extracted phrases and their associated phrase qualities. Phrase quality ranges from 0.0 to 1.0, with 1.0 being the highest quality. Alongside the outputted phrases, AutoPhrase also outputs other files, one being the Segmentation Model. This model file can be used for AutoPhrase's second function, phrasal segmentation. Using the model file, it will modify an input .txt file by marking identified phrases with phrase markers. This allows for further analysis as we can see the extracted phrases in each paper in the dataset, which allows us to count the frequency of phrases.

As mentioned previously, we created a function to process the DBLP v10 dataset. It aggregates the titles and abstracts of papers together in .txt files by the specified year ranges (intervals of 5 years). However, when running the phrase mining step of AutoPhrase, it does require sufficient training data, meaning that if the input .txt file is too small, the results will be mostly incoherent. We found that the minimum file size is around 200-300 kilobytes, but it is not always consistent, as some smaller files were able to run without errors. Regardless, it is better to have larger files. This is partially why we decided to group years together when looking at the dataset over time, as many of the earlier years did not have sufficient text data for the phrase mining step, as we can see from the distribution in Figure 1. Ultimately, grouping the years in intervals of 5 prevents this issue with the phrase mining step, and also allows for us to see more significant changes between each group.

3 Results

3.1 Phrase mining results

Table 1: AutoPhrase results on 1995-1999.txt

Phrase Quality	Phrase
0.9640877563	machine learning
0.9627931557	load balancing
0.9619412536	temporal logic
0.9618132944	dynamic programming
0.9615367883	sequent calculus
0.9604384138	resource management
0.9601422548	vector quantization vq
0.9598612067	reverse engineering
0.9595359994	gaussian elimination
0.9592247310	knowledge representation
0.9584704577	fuzzy logic
0.9580746756	normal form
0.9580186262	augmented reality
0.9579114197	pattern recognition
...	...

When running AutoPhrase on a single year range's .txt (containing all of its papers' titles + abstracts), we get an output of the phrases, along with their associated phrase qualities. Phrase quality ranges from 0.0-1.0, where 1.0 is the highest quality. We can typically associate high-quality phrases with single-word phrases with a score above 0.8 and multi-word phrases with a score above 0.5. These phrases provide insight into the various topics covered in just a single year range of published Computer Science papers.

This phrase mining step was run on each year range in the DBLP v10 dataset. The reason that we separated the year ranges into their own .txt files was that the extracted phrases do not have a year or year range associated with them. If we were to run the phrase mining step on the entire DBLP dataset, it would require processing the entire input dataset of titles and abstracts, and phrase matching with the phrase mining results to see which phrase belongs to which year range. Separating the papers by year range allows us to know which phrase and phrase quality is associated with a specific year range.

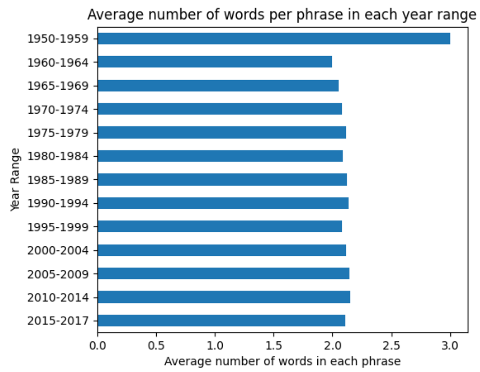

Figure 2: Average phrase length in each year range

After processing the phrase mining results, we calculated the average phrase length across each year range. The phrase length referring to the number of words in a phrase. The average phrase length is roughly two words long for each year range, with the exception of the earliest year range, 1950-1959.

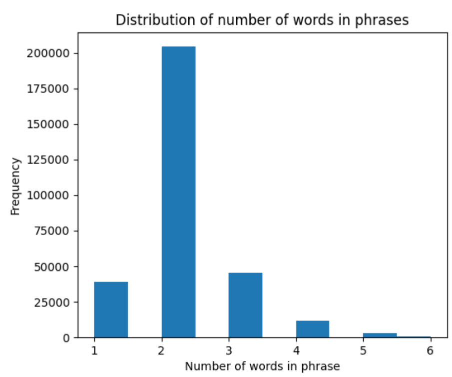

Figure 3: Distribution of phrase lengths across the entire DBLP dataset

This histogram shows the distribution of phrase length across the entire DBLP dataset. We can see that two-word phrases, or bi-grams, are the most common.

3.2 Phrasal segmentation results

After running the phrase mining step on each of the year ranges, the phrasal segmentation step was also run, using each year range's associated segmentation model from the phrase mining output.

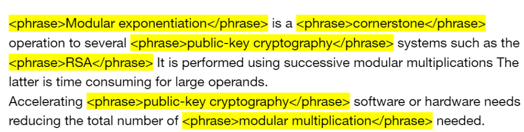
<phrase>Modular exponentiation</phrase> is a <phrase>cornerstone</phrase> operation to several <phrase>public-key cryptography</phrase> systems such as the <phrase>RSA</phrase> It is performed using successive modular multiplications The latter is time consuming for large operands.
Accelerating <phrase>public-key cryptography</phrase> software or hardware needs reducing the total number of <phrase>modular multiplication</phrase> needed.

Figure 4: Example phrasal segmentation results

The figure above shows an example of what phrasal segmentation does to text data. Any mined phrases with be marked with phrase markers. The phrase markers and phrases are highlighted in this screenshot for clarity. By processing the phrasal segmentation results, we can extract the marked phrases and group them together. This allows us to see the phrases mined by AutoPhrase on a per-paper level. For instance, with the example in the figure, if we consider it the text for a single paper, we can see that it contains the phrases: modular exponentiation, cornerstone, public-key cryptography, and RSA.

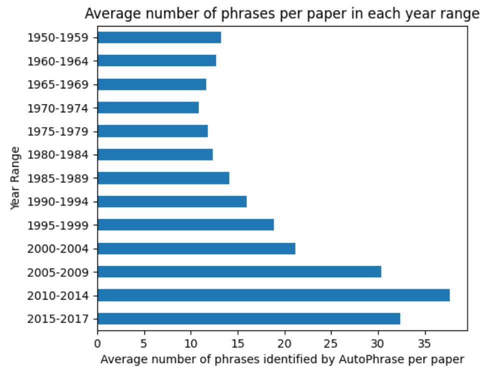

Figure 5: Bar chart of average phrases identified over time

This chart shows the average number of phrases identified by AutoPhrase for each year range. From processing the phrasal segmentation results, we are able to identify the phrases contained in each paper in the dataset. We can then calculate the average number of phrases identified across all papers in each year range, and then graph that information.

Here, we can see that the average number of phrases identified per paper generally increases over time. This can be due to factors such as average length of input papers for that year range, but could also be dependent upon the range of phrases displayed within a year range. A year range with more phrase variety could have less phrases show up per paper due to the lower average scores of the phrases causing them to be excluded from our high-quality phrase list.

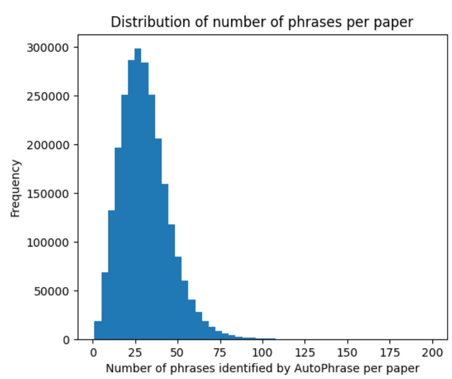

Figure 6: Histogram of number of phrases identified across entire dataset

This histogram shows the distribution of the number of phrases identified across the entire DBLP dataset. Overall, the number of phrases in a paper can vary widely, but the vast majority lie between 15 and 50 phrases.

3.3 Highest quality phrases over time

Table 2: Top 10 quality phrases across year ranges

1950-1959	1960-1964	1965-1969	1970-1974	1975-1979	1980-1984	1985-1989	1990-1994	1995-1999	2000-2004	2005-2009	2010-2014	2015-2017
operations research	tunnel diode	information retrieval	dynamic programming	fault tolerant	fault tolerance	logic programming	image segmentation	machine learning	heat transfer	block ciphers	option pricing	home automation
memory	differential equations	turing machine	markov chain	predicate calculus	human factors	pattern recognition	resource allocation	load balancing	belief propagation	microphone array	blind deconvolution	option pricing
magnetic	high speed	integer programming	question answering	linear programming	packet switching	petri net	petri net	temporal logic	stock market	hamming distance	laser scanner	rician fading
binary	data processing	data processing	programming languages	image processing	knowledge base	shortest path	character recognition	dynamic programming	congestion avoidance	wiener filter	superposition coding	voltage regulator
data	retrieval	automata theory	computational complexity	structured programming	dynamic programming	user interface	transaction processing	sequent calculus	kalman filters	copyright protection	moral hazard	buck converter
high	tunnel	dynamic programming	information retrieval	floating point	virtual memory	neural network	virtual reality	resource management	pattern recognition	blood pressure	brightness temperature	cooperative jamming
machine	amplifier	boolean function	feature extraction	question answering	turing machines	path planning	fourier transform	vector quantization vq	hamming distance	cellular automata ca	persistent homology	molecular docking
model	modulation	partial differential equations	floating point	dynamic programming	markov chain	load balancing	deductive databases	reverse engineering	random walks	transitive closure	associative memories	viral marketing
rate	digital	context free	integer programming	feature extraction	knowledge representation	image processing	modal logic	gaussian elimination	cellular phone	life sciences	buck converter	semidefinite relaxation
probability	design	differential equations	fault tolerant	transitive closure	petri nets	relational algebra	information retrieval	knowledge representation	stream cipher	spectral subtraction	preventive maintenance	mutual exclusion

We looked at the top 10 quality phrases across each year range to see how the phrase mining results differ across years. Taking a glance at these example phrases will help us determine if the quality phrases would serve as good predictors of a year. What is immediately obvious is that the first category consisting of papers with years from 1950-1959 consists of much simpler phrases. This category has the most single word phrases in their top 10 and their phrases illustrate broad concepts in Computer Science. This is promising as early Computer Science papers would deal with more basic concepts and could be a good predictor of year. This trend is relatively followed as the earlier year ranges contain phrases essential to the basics of Computer Science such as 'data processing' and 'information retrieval,' while papers in later years contain more high-level concepts such as 'vector quantization' and more proper nouns like 'Rician fading'.

There are some other aspects that stand out when looking at table 8. The phrase 'dynamic programming' appears in the top 10 of many year groups along with other phrases like information retrieval and feature extraction. The fact that AutoPhrase picks many high quality phrases that are not useful for discriminating year groups could lead to AutoPhrase's quality phrases being noisy data when trying to use for prediction. Another interesting factor is that the year category 2005-2009 contains a variety of phrases relating to biology such as 'cellular automata,' 'life sciences,' and 'blood pressure'. This could possibly be due to Computer Science as a field expanding into other disciplines once the foundations had been established. This could explain the appearance of many seemingly random phrases within later years that appear to have very little to do directly with Computer Science.

3.4 Most popular phrases over time

Table 3: Most popular multi-word phrases across year ranges

1950-1959	1960-1964	1965-1969	1970-1974	1975-1979	1980-1984	1985-1989	1990-1994	1995-1999	2000-2004	2005-2009	2010-2014	2015-2017
operations research (82)	pattern recognition (27)	sequential machines (85)	pattern recognition (165)	natural language (253)	natural language (494)	expert systems (782)	neural network (2504)	neural network (4977)	neural network (6001)	web services (12672)	cloud computing (16170)	machine learning (11254)
gaussian noise (16)	regular expressions (22)	pattern recognition (75)	linear programming (122)	pattern recognition (132)	signal processing (268)	natural language (770)	natural language (1089)	genetic algorithm (1700)	data mining (4901)	neural network (12314)	machine learning (14046)	big data (10885)
differential equation (12)	differential equations (21)	linear programming (71)	sequential machines (82)	computer graphics (128)	dynamic programming (204)	programming language (509)	expert systems (832)	image processing (1663)	web services (3543)	data mining (9980)	wireless sensor networks (12345)	social media (9504)
dynamic programming (8)	linear programming (19)	analog computer (58)	computer graphics (72)	linear programming (106)	pattern recognition (192)	user interface (495)	image processing (827)	software engineering (1430)	software engineering (3188)	wireless sensor networks (9382)	neural network (11381)	cloud computing (8373)
standard model (8)	sequential circuits (15)	sequential machine (54)	dynamic programming (69)	problem solving (104)	linear programming (174)	artificial intelligence (398)	distributed systems (799)	distributed systems (1414)	genetic algorithm (3115)	genetic algorithm (8088)	data mining (11235)	power consumption (6124)

By processing the phrasal segmentation results, we can obtain the frequency of each phrase in each year range's text. We specifically focused on the most frequent multi-word phrases across each year range in order to identify the most popular Computer Science topics in each period. We can see how the frequency of the top 5 phrases increases greatly over time, as more papers are published and topics of papers overlap. In the early years, there is a large focus on 'pattern recognition,' as it is in the top 5 in all of the year ranges from 1960-1984. Over time, this changes, with topics such as 'neural networks' and 'machine learning' becoming more prominent. Ultimately, this table provides insight into the most frequent phrases across each year range, and it does reflect the changes in the field as it has matured.

3.5 Phrase network visualization

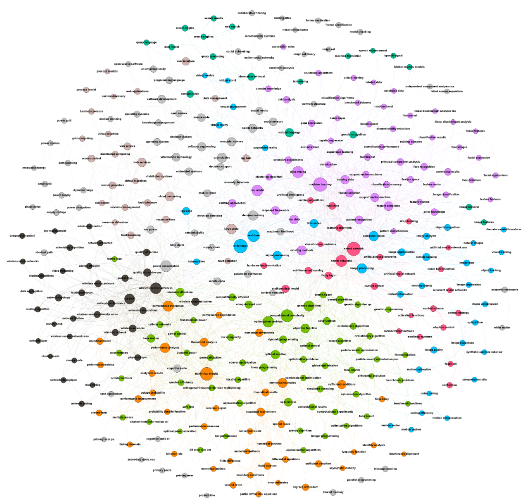

Figure 7: Network visualization

Higher-quality, zoomable image can be found here. This graph was created using the Gephi application after processing AutoPhrase's phrasal segmentation results on the DBLP v10 dataset.

This network visualizes the relationship between phrases for all papers in the DBLP v10 dataset (across all years). Phrases with more more occurrences in the dataset are represented by larger nodes in the network. Nodes are connected based on their connections in the paper. The phrasal segmentation results allowed us to extract the phrases identified for each individual paper in the dataset. With this, we could calculate the number of connections each phrase had with each other. For example, if 'neural network' and 'machine learning' are in the same paper, we would count that as 1 connection. With more connections across papers, edges between nodes have a larger weight.

Node colors are determined by modularity, so nodes with stronger edges to each other will be grouped together. For instance, with the purple nodes, 'machine learning' is the largest node, and we see other related nodes to that topic, such as 'decision trees', 'support vector machines', etc.

We only included multi-word phrases in the network, with a minimum threshold of 150 for the edge weights. This means that only phrases with at least 150 connections to each other are included in the network. This threshold is necessary as there are so many phrases and connections within the entire DBLP dataset. It allows us to visualize the relationships between the most frequent and most commonly occurring phrases.

3.6 Phrase network by year range

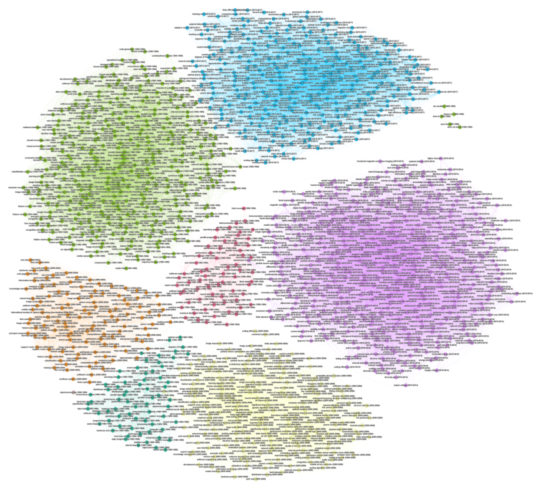

Figure 8: Yearly network visualization

Higher-quality, zoomable image can be found here.

This network isolates phrase relationships to their year range, providing insight into the most popular and connected phrases in each year range. The nodes colors are based solely on the year range of the phrase, rather than modularity. One fact to take into account is that the number of papers is much higher is recent years, so the frequency of phrases and their connections is much higher compared to earlier years. Steps were taken to normalize this difference across each year range and to only display the strongest and most meaningful relationships, but the number of nodes for each year range is not exactly equal. Ultimately, the purpose of this network is to provide a more intuitive understanding of phrase connections in relation to time.

3.7 Classification model

One of our initial goals for this project was to create a classifier to predict the year of a random Computer Science paper in order to demonstrate how distinct phrases contained within certain years have the capability to identify what year of the input paper. For this, we attempted multiple types of models including a Jaccard-based predictor, a predictor using phrase overlap between years, as well as trained models using one-hot encoding. We were able to successfully create a model using a combination of the TF-IDF (Term Frequency-Inverse Document Frequency) text-vectorization and grouping of multiple years. We were able to achieve a 0.79 f1 score on the test set.

Figure 1 (Document count for DBLP v10) shows the imbalance of paper count per year. It is not feasible to predict a random paper up to the accuracy of a year. To mitigate the imbalance of the paper count distribution, we grouped the papers into several-year brackets as shown in Table 9. The "integer encoding" simplified the coding.

For each paper, we used the high-quality phrases extracted by AutoPhrase from the abstract and title of the paper. We filtered out some high scoring irrelevant phrases such as, "paper argues", "paper considers", and etc. This was done by generating our own stop-word list of the irrelevant phrases by reviewing the extracted high-quality phrases. Afterwards, we converted the high quality phrases using TF-IDF text-vectorization changing the phrases into a fixed-length feature vector. We decided to consider the top 1000000 phrases ordered by term frequency across all of the papers when building the vocabulary. Our first baseline classifier utilized One-vs-the-rest (OvR) multi-class strategy to classify a paper into the year brackets. Due to the imbalance of paper count distribution, we used StratifiedShuffleSplit to perform a train-test split following the distribution of "year-bracket" so that the train dataset and test dataset preserve the same distribution. Our baseline classifier resulted in a 0.77 f1 score. By comparing the classification performance of different classifiers such as

year-bracket	Encoded	Paper#(%)
1950-1959	0	0.0131
1960-1964	1	0.0334
1965-1969	2	0.1037
1970-1974	3	0.2185
1975-1979	4	0.3677
1980-1984	5	0.6610
1985-1989	6	1.2859
1990-1994	7	2.8021
1995-1999	8	5.6662
2000-2004	9	12.1526
2005-2009	10	25.5802
2010-2014	11	34.2395
2015-2017	12	16.8761

Table 4: Year bracket partition, integer encoding, and paper count distribution

LogisticRegression and svm.LinearSVC, we found that svm.LinearSVC had the best performance. We then used GridSearchCV method to search for the best C hyper-parameter of svm.LinearSVC.

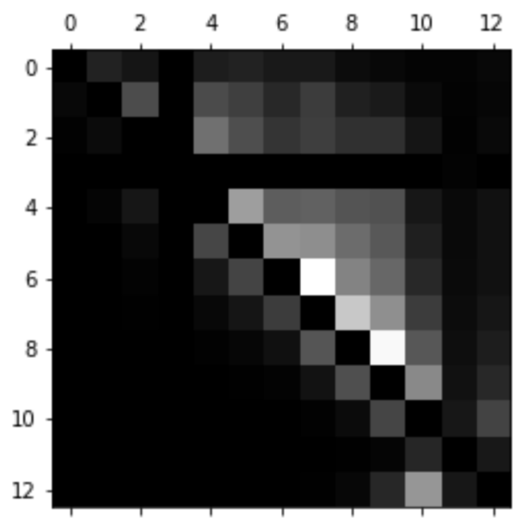

Figure 9: Normalized Confusion Matrix

This confusion matrix was obtained by analyzing our final model utilizing svm.LinearSVC. The normalized confusion matrix shows our model tends to predict the year later than the actual year. The imbalance of the paper count (the later year has much more papers) is causing the issue. We need to further mitigate the imbalance of the paper-per-year count.

We did not use the position info of the phrases when we performed the TF-IDF text-vectorization. The "ngram_range" parameter of TfidfVectorizer can be used to catch the position info of the multiple phrases.

4 Conclusion

After processing and exploring the DBLP v10 dataset, we were able to utilize both functions of AutoPhrase (phrase mining and phrasal segmentation) to extract meaningful data and explore the relationships between phrases further. We identified the change in phrases over time by looking at the most popular phrases for each year range. We analyzed the relationship between phrases on a per-paper level, utilizing the segmentation results, in order to create a network visualization. We analyzed this relationship with respect to time, visualizing the network of phrases for each year range. We created a classification model in order to predict the year range of a paper based on its phrases.

5 Future Work

Additional work can be done to improve the classification model idea. The proposed idea was to be able to pass in any random input paper title and abstract and obtain a prediction of the specific year. Perhaps by utilizing the phrasal segmentation results and additional features to train a model, it may be possible to revert back to making predictions to specific years, rather than the defined year ranges.

It would be interesting to explore an evolving network animation that starts with the first year in the dataset, showing all of the phrase relationships, then dynamically changes as we go through each year. This may not be possible directly in Gephi's software, but it could be done by creating separate graphs and maintaining certain color schemes. Additionally, exploring single-word phrases alongside the multi-word phrases could be interesting as well. It may require additional filtering of words to remove any meaningless phrases.

A Appendix: Phrase matching and phrase similarity

Note: This analysis was done prior to our decision to group papers by 5-year ranges, so it examines the phrase mining results on a per-year basis.

Table 5: Direct phrase matching for 'convolutional neural networks'

Phrase Quality	Phrase	Year
0.865809	convolutional neural networks	2012
0.915629	convolutional neural networks	2013
0.937014	convolutional neural networks	2014
0.931728	convolutional neural networks	2015
0.917273	convolutional neural networks	2016
0.904261	convolutional neural networks	2017

Table 6: Phrase similarity for 'convolutional neural networks' (Using unique phrases overall)

Phrase Quality	Phrase	Year	Distance
0.865809	convolutional neural networks	2012	0.0
0.900172	convolutional neural network	2013	1.0
0.839879	convolution neural network	2016	3.0
0.918423	convolutional neural networks cnn	2015	4.0
0.915458	convolutional neural network cnn	2014	4.0
0.889687	deep convolutional neural networks	2014	5.0
...	...	...	...

The phrase mining results per year are aggregated into a single .csv file with a new column containing the phrase's year. It is possible for multiple instances of a phrase to appear in the file, as they will have a different associated year and generally have a different phrase quality value. We can utilize Pandas to read in this file and perform various operations. For example, when looking at the value counts of the phrases, we can see popular phrases that show up many times, such as 'natural language,' 'data structures,' and 'artificial intelligence.' We can then check for direct matches of a phrase, such as checking the rows that have the phrase 'image processing'. When doing so, the phrase first appears in our dataset in 1981 and has appeared in every year since, all the way until 2017.

Although phrase matching allows for us to directly find a phrase and the years in which it appears, it does not account for potential misspelling or non-direct matches. For example, if we tried to match for 'convolutional neural networks' but the dataset only contained 'convolutional neural network' (not plural).

We utilized the Levenshtein package to measure the Levenshtein distance between strings. This allows for us to find phrases in the dataframe that may not be exact matches, but are similar enough to warrant further analysis. When looking for the phrase 'convolutional neural networks', there is a direct match in the dataframe, but there are also other phrases that are extremely similar, such as 'convolutional neural network' and 'convolutional networks'. This approach looking at phrase similarity allows for us to find the most similar phrases to the input phrase, without having to worry about having a direct match in the dataframe. We believe this idea can be utilized to consolidate phrases within the phrase mining results, as there are commonly multiple instances of extremely similar phrases, such as 'neural network' and 'neural networks.'

This could also be used as an alternative method to classify an input paper's year, or to provide information on the various phrases within an input paper and the years in which those phrases originate. For example, if we take in a paper's title and abstract, we can extract the phrases within it, by using an n-gram model, or the phrasal segmentation function. Then we could use phrase similarity to find the similar phrases. So, if a paper contained the phrase 'convolutional neural networks,' we could say that we found a match of that phrase, and that it first appeared in 2012, with continual appearances until 2017.