NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

DEPARTMENT OF
COMPUTER SCIENCE

GONÇALO ABRANTES FAZENDA

BSc in Computer Science

ATTRIBUTE SELECTION FOR UNSUPERVISED AND LANGUAGE INDEPENDENT CLASSIFICATION OF DOCUMENTS

SELECTION AND EXTRACTION FROM LANGUAGE INDEPENDENT *CORPORA* IN A TEXT-MINING CONTEXT

MASTER IN COMPUTER SCIENCE

NOVA University Lisbon
December, 2022

DEPARTMENT OF
COMPUTER SCIENCE

ATTRIBUTE SELECTION FOR UNSUPERVISED AND LANGUAGE INDEPENDENT CLASSIFICATION OF DOCUMENTS

SELECTION AND EXTRACTION FROM LANGUAGE INDEPENDENT *CORPORA* IN A TEXT-MINING CONTEXT

GONÇALO ABRANTES FAZENDA

BSc in Computer Science

Adviser: Joaquim Francisco Ferreira da Silva
Assistant Professor, NOVA University Lisbon

Examination Committee

Chair: António Maria Lobo César Alarcão Ravara
Associate Professor, FCT-NOVA

Rapporteur: Vitor Rocio
Associate Professor, Universidade Aberta

Adviser: Joaquim Francisco Ferreira da Silva
Assistant Professor, FCT-NOVA

MASTER IN COMPUTER SCIENCE

NOVA University Lisbon
December, 2022

Attribute Selection for Unsupervised and Language Independent Classification of Documents

This document was created with the (pdf/Xe/Lua)LaTeX processor and the NOVAthesis template (v6.8.4) [1].

Scientia potentia est.
Ex nihilo nihil fit.

Acknowledgements

During these past years of my academic endeavours, I have learned a great deal and grown as a person, made friendships that I hope last for a lifetime, and tackled many obstacles. Certainly, I could not have done it without the help and support of various people and entities, to each I express my sincere thanks:

To my Advisor, Professor Joaquim Silva, thank you for always being present whenever I needed assistance and always being ready to answer any question, no matter how many times it needed explaining.

To our amazing institution, the NOVA School of Science and Technology, thank you for all the opportunities, knowledge, education, and friends I have met along these past years.

I would like to extend a personal thanks to the following friends, who I hope will accompany me throughout life, through good times and bad. To my university *"besties"*: Beatriz Rebelo, Sara Simões, Bernardo 'Boris' Baldaia, Eduardo Subtil, Luís Coelho and Ricardo Leitão, thank you for all the good moments, inside-jokes, and hangouts over the years. You've made this stage of my life much more fun and it definitely wouldn't be the same without you all. And to my old buddies: Sara Vieira, Ricardo 'RK' Fortes, Rúben Gaspar, Bruno Marques, Rafaela Reis, and Mariana Almeida, thank you for everything.

To Ana Piedade, thank you for always being there for me, for your unconditional love and support, and for being my inspiration. You bring out the best of me and I wouldn't trade our moments together for anything else; I love you.

My thanks to my grandparents, for their abundant love throughout the years, my cousins for the hangouts and jam sessions, and my aunt and uncle for always supporting me.

Finally, yet just as important, to my parents, Rui and Susana, and my sister Joana, for always being there and looking out for me, for being my safety net when times are rough, and for everything they've done and sacrificed for me. To both my cats, Luke and Barry, thank you for meowing at four in the morning to remind me to go to bed.

“Labor omnia vincit ” (Virgil)

Abstract

Raw text documents are the most common way documents are written, that is, unstructured text. So, they contain most of the information available. Thus, it is desirable that there are tools capable of extracting the core content of each document and, through it, identify the group to which it belongs, since in unstructured texts there is usually no foreseen place for indicating the document class. Nowadays, English is not the only language documents appear in the available repositories. This suggests the construction of tools that, if possible, do not depend on the language in which the texts are written, which is a challenge.

This dissertation focuses mainly on clustering documents according to their content, using no class labels, that is, unsupervised clustering. It aims to mine and to create features from text in order to achieve that purpose. It is also intended to classify new documents, in a supervised approach, according to the classes identified in the unsupervised training phase.

In order to solve this, the proposed solution finds the best features inside the documents, and uses their discriminative power to provide clustering. In order to summarise the core content of each cluster found by this approach, key expressions are automatically extracted from their documents.

Keywords: Information Retrieval Systems, Natural Language Processing, Feature Selection and Extraction, Text Mining, Document Classification, Document Clustering

Resumo

Documentos de texto bruto são a forma mais comum de escrita de documentos, ou seja, texto não estruturado. Assim, eles contêm a maioria das informações disponíveis. Deste modo, é desejável que existam ferramentas capazes de extrair o conteúdo mais importante de um documento e, por este meio, identificar o grupo ao qual o documento pertence, pois em textos não estruturados geralmente não há uma previsão de indicação da classe do mesmo. Atualmente, o Inglês não é a única linguagem em que os documentos aparecem nos repositórios disponíveis. Isto sugere a construção de ferramentas que, se possível, não dependam da linguagem em que os textos são escritos, sendo isto um desafio.

Esta dissertação foca-se principalmente em agrupar os documentos de acordo com o seu conteúdo, sem usar rótulos de classes, ou seja, agrupamento não supervisionado. O objetivo será alcançado através da extração e criação de atributos a partir do texto. Pretende-se também classificar novos documentos, numa abordagem supervisionada, de acordo com as classes identificadas na fase de treino não supervisionado.

De modo a tentar resolver este problema, é proposta uma solução que encontra os melhores atributos nos documentos, e usa o poder discriminativo das mesmas para fazer o agrupamento. De modo a sumarizar o conteúdo principal destes agrupamentos, expressões chave são automaticamente extraídas dos documentos.

Palavras-chave: Sistemas de Extração de Informação, Processamento de Línguagem Natural, Seleção e Extração de Atributos, Mineração de Texto, Classificação de Documentos, Agrupamento de Documentos

Contents

Appendices

List of Figures

Glossary

algorithm A set of well defined and finite rules to be executed by a computer.

benchmark The act of running a computer program or other operations, for the purpose of assessing the relative performance of an object.

corpus A collection of written or spoken material stored on a computer.

cross-validation A method that uses different folds of the data to test and train a model.

feature In machine learning, a feature is an independent, measurable characteristic of a data object.

fold A split of the data into training and test sets, used during the various iterations of cross-validation.

hyperparameter Parameters that are manually set, that are used to control the training/learning process of a model.

inflection In linguistic morphology, it is the process of word formation in which a word is modified to express different grammatical categories.

library A collection of resources used by a program or software.

linear transformation A function from one vector space to another, that respects the structure of each vector space.

meronym Term that denotes part of something, but refers to the whole of said part.

meta-class Document classes unknown by the system, but known by the developers.

metric Measurement of characteristics that are quantifiable or countable, that help evaluate results.

parsing Dividing a string into its singular components.

production The final stage of software development.

Python A high-level, general-purpose programming language.

string A sequence of characters.

vector space A group of vectors, added collectively and multiplied by scalars.

XML A markup language and file format for storing, transmitting, and reconstructing data.

Acronyms

ACC	Adaptive Classifier Combination
BIRCH	Balanced Iterative Reducing and Clustering using Hierarchies
CBOW	Continuous Bag-of-Words
CF	Clustering Feature
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
EM	Expectation Maximization
GMM	Gaussian Mixture Model
HDBSCAN	Hierarchical DBSCAN
IR	Information Retrieval
LDA	Latent Dirichlet Allocation
LLSF	Linear Least Squares Fit
MWU	Multiword Lexical Unit
NLTK	Natural Language Toolkit
PCA	Principal Component Analysis
SCP	Symmetrical Conditional Probability
SI	Specific Mutual Information

SVD	Singular Value Decomposition
SVM	Support Vector Machines
WWW	World Wide Web

1

Introduction

This chapter serves as an introduction and contextualization of the problem at hand, as well as motivations and expected contributions of this dissertation to said problem.

1.1 Context

In the recent years, there has been a booming growth of online text libraries and documentation, as well as raw sources of data that often need to be categorized so that they may be organized more easily. As such, there has been an increase in the concern of having robust and reliable unsupervised text labelling and categorization systems in an Information Retrieval (IR) context, as these systems allow us to more easily find interesting information on the World Wide Web (WWW) that arises everyday, and classify them accurately.

But, due to the dynamic nature of these sources, it is much more difficult to cluster and label these sources correctly within a limited set of options than otherwise anticipated. Another problem arises when we're dealing with multi-language data sources as they introduce a new layer of abstraction where the metrics used for phrasing and word extraction may not work for certain languages (take for example Russian language with Cyrillic alphabet versus Portuguese language with Latin Alphabet in a *corpus*).

In the past, researchers have tried to use machine learning approaches [2], but these approaches often used either supervised or semi-supervised techniques. One example of a supervised approach would be the usage of a pre-labeled set of documents for training data, with which the classifier would be trained with. With this approach, apart from needing to be manually labeled, the labels are somewhat static and limited to previously labeled, that is, the system does not have the capability of learning new classes. New classes will only be learned with the aid of human labelling.

1.2 Motivation and challenges

Using supervised approaches may yield some very positive results [3] but, as mentioned previously, they require some prep-work done before the classifiers can be used. Knowing which documents belong to which class is already something very useful in training the text classifiers but the lack of this information poses a challenge in unsupervised learning, where the classifier does not previously know which category those documents belong to. This lack of support suggests that there are expressions in text that, by having strong semantic meaning, must not be ignored in order to build a possible set of features to discriminate document classes.

Some semi-unsupervised classification techniques also began surfacing [4], where some labels would be extrapolated from previously known labelled documents, ultimately having the same problem as supervised approaches.

Hence grows the motivation to create a fully autonomous unsupervised classifier in order to classify and organize massive amounts of data into more easily distinguishable clusters, for easier retrieval of information.

The unsupervised categorization of the documents is a difficult task because there is a need to create new features capable of capturing the "essence" of a category from written text. In fact, there are groups of words in text that may allow for easier categorization of text, for example: if "economic crisis" is found in text it is quite likely that this text document can be categorized as "economy" or "finances". On the other hand, if we find an expression like "at this moment" we do not expect this expression to better help correctly categorize the document.

Another motivation is that, once the clusters are built through unsupervised categorization, these may be used to classify new entry documents as if we were in a supervised classification context.

For a classifier to correctly identify the labels of each document, the features need to be very carefully selected from the *corpus* and, since we want language independent classification, these discriminating features must be even more judiciously selected. The goal and main motivation of this dissertation is to carefully select the best features, and mine labelling categories in a given *corpus*.

As such, several metrics and approaches will be taken into account when extracting the features. Words and sentences will need to be transformed into suitable representations to be given to the classifiers (for example, separating punctuation and other special characters in text), but their meaning should not be altered. An example of this would be in the sentence "Maria, João, Almeida"; if, by chance, we removed the commas from the sentence, we would have a valid Portuguese compound name – "Maria João Almeida" – instead of an enumeration consisting of 3 different people, thus completely altering the meaning of the sentence.

Features in text classification are very abundant, since we may look at every word as a potential feature and, as such, every word must then be accounted for as a potential source of information. This in turn creates a very large feature space, which implies a feature reduction process that must be elaborated in order to reduce them to a manageable size.

Another challenge is to be able to extract the core content of each cluster that were built in the unsupervised learning phase, in order to be understood by users.

1.3 Contribution

In this dissertation, the following objectives were achieved:

- To build an unsupervised approach capable of clustering documents according to their categories. For that, appropriate features had to be created in order to mine similarities between documents of the same class and dissimilarities between documents of different ones.
- To keep language independence. For that, specific morphosyntactic information and and other language dependent tools were avoided in the development of the approach.
- To classify new documents. Once clusters are created with high enough precision in the unsupervised phase, new document samples can be classified accordingly, now that groups/classes were found.
- To extract the main contents of the clusters. Once documents are grouped in the unsupervised phase, the main content of each cluster can now be automatically extracted.

1.4 Structure

Besides the introductory chapter, this document is comprised of the other following chapters:

- **Chapter 2** - Background and state of the art. This chapter aims to define some very important concepts related to this dissertation. Furthermore, it will also explore and discuss results obtained from other researchers that used other tools and techniques in this field.
- **Chapter 3** - Proposed solution. This section presents our implemented solution and explains the techniques used in further detail.
- **Chapter 4** - Results. Results obtained during testing and experimentation regarding different metrics will be outlined here.
- **Chapter 5** - Conclusion. The final chapter will review the work that was done, and present possible improvements for future work.

2

Background and state of the art

This chapter serves as an small introduction and overview to previous work done in this field, from important topics and concepts to tools and techniques, used by previous researchers.

2.1 Overview

The whole process of building and training a system to correctly classify and cluster text documents is quite complex, usually divided into several major phases [3, 5], which will all be explained succinctly and whose techniques and tools will be detailed in the following sections. It is worth noting that, due to the vast amount of available techniques, certain ones will not be explained due to its specificity or sparse usage by researchers in this field, in favor of more used tools.

At the end of this chapter, and in Sec. 2.6, we will discuss empiric results of tests conducted on the more prevalent techniques.

2.2 Preprocessing phase and techniques

Firstly we need to select the necessary features from raw text. To do so, it is necessary to preprocess the text into more relevant information to then extract the features from. This phase is usually comprised of several steps that may change depending on how the text is intended to be used, and how the text appears in the *corpus*. For example, removal of XML tags may be optional in some cases, while in others these may be used as source of information [6].

2.2.1 Stemming

Suffix Stripping or Stemming [7] is an useful technique in Information Retrieval (IR) systems as it allows a reduction of a group of terms into a single term – the *stem*. Take the term “Wander” for example; It may assume many forms, be it a noun “Wanderer”, the past participle “Wandered” or another possible conjugation “Wandering” (Present Continuous), but it may also be condensed into a single term – “Wander”. What this technique aims to do, is to remove suffixes in order to return the word to its *stem* form, as to reduce the amount of different variations of the same word into a more digestible size for the system to work with. There are however several stemming algorithms, each with their own differentiated outputs which may or may not be more useful in some cases [8].

2.2.2 Lemmatization

On the same topic as stemming, there also exists Lemmatization [9], which is the process of grouping inflected forms of certain terms, allowing them to be reduced to a single term as well. This differs from stemming as the resulting singular term is the basis of all its inflected parts – stemming a word may not result in a morphological correct word – for example, the removal of the suffix “ed” through stemming in the word “tied”, results in “ti” but lemmatizing it would result in “tie”. Lemmatization offers a more morphologically correct representation of the lemmatized word, but it is more computationally intensive than Stemming.

2.2.3 Tokenization, punctuation, digits, and stop-word removal

Tokenization is the process of parsing the text into tokens. The resulting tokens are then used in the remaining procedures.

Other preprocessing techniques usually include punctuation, digits, and stop-words removal [3, 10], as according to some authors, these rarely discriminate possible features in a document. Expressions and terms such as “the”, “and” and “but” are good examples of stop-words. As for punctuation, and as mentioned previously in Ch. 1, completely removing it may strongly alter the meaning of a sentence, and we want to avoid it as much as possible. In most cases, digits are ignored due to their weak discriminant power, though sometimes they may be important.

2.3 Feature selection and extraction techniques

Feature selection is very important, as the central premise of this selection is to eliminate or discard redundant and irrelevant features. Primarily in text classification, there are quite an abundance of features that are not at all discriminating of what we exactly need. For example, and as mentioned previously in Sec. 1.2, certain words are of no use in document category discrimination, while others may be of very valuable use.

It is also worth noting the difference between “feature selection” from “feature extraction”. The fundamental difference is that feature extraction creates new features from the original ones, whereas feature selection simply selects a subset of the original features.

After obtaining the most relevant features, it may be necessary to reduce their dimensions to a more manageable size to input to the classifiers, without sacrificing a lot of classification accuracy and retaining as much possible variance contained in the original features. To do so, Principal Component Analysis (PCA) may be employed. PCA works by computing new principal components, which are linear combinations of the initial variables, by combining them in a way that most of the information is set in the first few components – meaning that even if we have a lot of components, only the first few will actually have meaningful data. These principal components represent data orientation with maximal amounts of variance, as the higher the variance, the higher the data point dispersion along those vectors. Often times, using PCA improves classification and clustering results [11].

2.3.1 WordNet

WordNet links words into semantic relations, such as synonyms, antonyms and meronyms in a lexical database [12]. The main interest in using WordNet is that it was built with the support for automatic text analysis and artificial intelligence in mind, as it improves the quality of resulting clustering due to semantic similarities. It works through grouping words into *synsets*, that each represent a lexical concept which can then be used to create features [5].

2.3.2 TF-IDF – Term Frequency-Inverse Document Frequency

This metric assesses how important a term t is in a document d. Although Term Frequency regularly suggests the absolute frequency of term t in d, lately this factor has been surpassed by the use of the relative frequency of t in d, $TF(t,d)$, in order to take into account the size of the document, thus normalizing the occurrence frequency of t.

$$TF(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \tag{2.1}$$

In the above equation, $f_{t,d}$ is the absolute frequency count of t in d whilst t' is any term occurring in document d. Although it is acceptable that the higher the relative frequency $TF(t,d)$ the more important t is in d, t will be more important in d, if t is rare in the other documents. So, the inverse of the regular document frequency (IDF) is a weight indicating in how many documents the word appears – the higher this number, the lower the value.

$$IDF(t) = \log \frac{|D|}{|\{d' \in D \wedge f_{t,d'} > 0\}|} \tag{2.2}$$

Therefore, $TF-IDF(t,d)$ is the resulting statistic of combining $TF(t,d)$ and $IDF(t)$. This is very valuable as it ensures that lower frequency terms are much more discriminating, as opposed to more regular ones like “the”, “and” and so forth, due to the normalization imposed by $IDF(t)$.

$$TF\text{–}IDF(t,d) = TF(t,d) \times IDF(t) \tag{2.3}$$

2.3.3 Term Contribution (TC) and Term Strength (TS)

Another feature selection method is Term Contribution, where the contribution of a term is measured by how it affects the the documents' similarity [13]. The similarity between two documents, d_1 and d_2 can be computed as such:

$$similarity(d_1, d_2) = \sum_t f(t, d_1) \times f(t, d_2) \tag{2.4}$$

Where $f(t, d_n)$ represents the $TF{-}IDF(t, d)$ statistic.

On the other hand, the overall contribution of a term to the similarities of the documents in the *corpus* can be computed as:

$$TC(t) = \sum_{i,j\,:\,i \neq j} f(t, d_i) \times f(t, d_j) \tag{2.5}$$

Term strength is a technique whose core idea is to measure how informative a term t is, relating two documents d_1 and d_2. It was defined in [14] as:

$$s(t) = P(t \in d_1 | t \in d_2), d_1, d_2 \in D \wedge similarity(d_1, d_2) > \beta \tag{2.6}$$

Where β is a threshold parameter to determine if the pairs are related.

2.3.4 Word2Vec and Doc2Vec

Word2Vec is an algorithm created by Tomas Mikolov et al. [15] that uses a Neural Network model, which is trained to learn association between words. The model, after training, is able to detect word correlations and synonyms. Training the can be done using two different architectures:

1. Continuous Bag-of-Words (CBOW), in which the model predicts the word from its surrounding words;

2. And Skip-Gram, where it weighs context words based on distance from current word.

Each word is represented as a vector, which are then further processed in order to find semantic similarities between those vector represented words.

The Doc2Vec is an extension to Word2Vec, done by T. Mikolov as well [16, 17], in which the Word2Vec architecture was extended by allowing the model to take a token of a document as input. By using the CBOW model with some slight additions to the way it works, Doc2Vec now, instead of exclusively using words to predict the next word, uses another feature vector unique to the document, that is trained alongside the word vectors and holds a numeric representation of the document by the end of its training.

2.3.5 Information Gain

Information Gain measures changes in entropy when a certain feature t is absent or present. In classification problems, Information Gain can be used to measure how common a feature is in a particular label when comparing to all other labels. For example, if the occurrence of the word “finances” in a *corpus* makes the entropy drop less than the term “association”, then “finances” is more qualified to use as a feature [18].

$$IG(t) = -\sum_{i=1}^{m} P_r(c_i) log P_r(c_i) + P_r(t) \sum_{i=1}^{m} P_r(c_i|t) log P_r(c_i|t) + P_r(\bar{t}) \sum_{i=1}^{m} P_r(c_i|\bar{t}) log P_r(c_i|\bar{t}) \quad (2.7)$$

Where $P_r(c_i)$ stands for the *a priori* probability of category/class c_i; $P_r(t)$ and $P_r(\bar{t})$ are the *a priori* probabilities of the presence and absence of t respectively.

2.3.6 Chi Squared χ^2

The χ^2 statistic compares the difference in measurement of the data towards the expected distribution, and so measures the lack of independence between a term t and a class/category c_i. If the term t and category c are independent, then the value of the χ^2 is zero [19]. A high value of χ^2 reflects strong dependence.

$$\chi^2(t, c_i) = \frac{N_d[P(t, c_i)P(\bar{t}, \bar{c_i}) - P(t, \bar{c_i})P(\bar{t}, c_i)]^2}{P(t)P(\bar{t})P(c_i)P(\bar{c_i})} \quad (2.8)$$

In the above equation, $P(t, c_i)$ denotes the probability of the feature/term t occurring in a document which belongs to category c_i. The constant N_d denotes the cardinality of the document set.

2.3.7 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) is a generative probabilistic model that is able to extract latent (or hidden) topics from a *corpus* [20]. Documents are viewed as a mixture of latent topics, which themselves are constructed from a mixture of the probability of words or phrases found inside the documents of the *corpus*.

LDA has two major hyperparameters – α and β. The α parameter controls the distribution (probabilities) of all topics to assign to each document. For example, a low value of α tends to assign a single topic to each document. The β parameter controls the distribution of different words or phrases to assign to each topic. Higher values of β tend to assign a more homogeneous mixture of words and phrases to each topic.

Another parameter that is also needed for LDA to function is the topics parameter, which means, the number of topics LDA extracts. One common drawback is that this number needs to be stated, which at times is not possible to do or hard to estimate.

2.3.8 N-grams and LocalMaxs

In computational linguistics, an n-gram is a sequence of n items from a sample of text. A Multiword Lexical Unit (MWU) is any string of text or speech that makes up for a compound nouns, adverbial and prepositional locutions, to name a few.

LocalMaxs is an algorithm that is able to extract MWUs from text, based on statistical calculations between n-grams. The core idea of the algorithm is that n-grams are held together by "glue" and different n-grams have different values of "glue" – for example a *bi*-gram compound noun has a much stronger glue than a *bi*-gram composed of a preposition and a verb, as prepositions and verbs tend to appear more often and in conjunction with other words, thus lowering the glue value.

In order for an n-gram to be classified as a MWU, its glue score must be a local maximum concerning its neighbourhood. To do so, we need the glue values of every $(n-1)$-gram contained in the current n-gram, and the glue values of every $(n+1)$-gram in which the current n-gram is contained. Let W be our current n-gram, $\Omega_{n-1}(W)$ be the set of all glues of the n-grams contained in W, and $\Omega_{n+1}(W)$ be the set of all glues of the n-grams that contain W. LocalMaxs states that, for a n-gram W, it is a MWU if and only if:

$$\begin{gathered}(length(W) > 2 \wedge freq(W) > 1 \wedge g(W) > \frac{max(\Omega_{n-1}(W)) + max(\Omega_{n+1}(W))}{2}) \\ \vee \\ (length(W) = 2 \wedge freq(W) > 1 \wedge g(W) > max(\Omega_{n+1}(W)))\end{gathered} \tag{2.9}$$

Where $g(W)$ stands for a generic function for measuring the glue of W. When W is composed by more than two words (2-gram), the n-gram W must be transformed in a pseudo 2-gram for obtaining a normalized glue value. For that, the $n-1$ dispersion points of W are considered by dividing the n-gram W in all different several left and right pair parts: for example the 5-gram "the sky is beautiful today" can be broken down into the following set of $(n-1)$ left-right pairs: { "the", "sky is beautiful today"}, {"the sky", "is beautiful today"}, {"the sky is", "beautiful today"}, {"the sky is beautiful", "today"}. Then, the pseudo 2-gram glue value will be the average glue values of the different 2-grams pairs.

There are various possible ways of calculating the "glue" values of n-grams, such as the *SCP* (Symmetrical Conditional Probability), the ϕ^2 Coefficient, the *Dice* Coefficient and *SI* (Specific Mutual Information) [21]. These metrics will be compared in Sec. 2.6.

2.4 Text and document classification algorithms and techniques

The process of classification corresponds to the usage of an algorithm to sort and label classes of information, being able to be performed in both types of data (structured or unstructured). It uses a function which is applied to the input of the classifier, that maps the objects to classify, to discrete output variables. In other words, the classifier model uses previously learnt knowledge to predict a possible label (in our case, category) for a document based on its features.

Classification problems may fall into several categories such as binary (for example Boolean classification - true or false), multi-class (each sample is only assigned to one and only one label) and multi-label (each sample is assigned to a group of labels). In document classification cases, albeit rare, it is possible that the document may be labelled as two or more different categories.

One possible way of measuring the performance of the classifiers is the use of Recall and Precision measurements of the model. Precision is the percentage of correctly predicted documents (True Positives) by the classifier, out of the total number of documents that it predicted for the label (True Positives plus False Positives), while Recall is the percentage of predicted documents of a label (True Positives) out of the total number of documents it should have predicted for that given label (True Positives plus False Negatives). Another possible measurement is the F-measure, in which the harmonic mean of Precision and Recall is calculated. Yet another measure for performance is the Accuracy which is calculated by dividing True Positives plus True Negative by the sum of True Positives, True Negatives, False Positives and False Negatives.

$$\begin{gathered} Recall = \frac{\text{labels found and correct}}{\text{total labels correct}} \\ Precision = \frac{\text{labels found and correct}}{\text{total labels found}} \end{gathered} \tag{2.10}$$

2.4.1 Naïve Bayes

The Naïve Bayes classifier is based on the Bayes' Theorem. It returns the class k which maximizes the sum of the logarithm of the *a priori* probability of the class k plus the sum of the logarithm of the conditional probability of each feature x_i, given the class k.

$$C = \underset{k \in \{0,1,\ldots,K\}}{argmax}\ ln\ P(C_k) + \sum_{i=1}^{N} ln(P(x_i|C_k)) \tag{2.11}$$

The classifier assumes that the different features are independent from each other within any class, meaning that features do not influence each other despite being present at the same time.

In most of the practical cases, complete feature independence does not occur, however, this classifier can be used to produce good results.

2.4.2 *K*-Nearest Neighbors

K-Nearest Neighbors is a classification algorithm in which it tries to predict a class of an element x by selecting the most common class of the K nearest points to x. In other words, the gist of this algorithm is that known data is arranged in a space defined by features and, when new data is given, it will compare the classes of the closest K-neighbors to determine the class of the new data. In text and document classification, K-Nearest Neighbors takes as input a document represented as a vector of word weights, outputting a list of categories with a confidence score for each of them [18].

Figure 2.1 shows a visual example of how the K-Nearest Neighbors algorithm works. Depending on the value of K given as a parameter, it calculates the distances to nearest points, selects the K nearest data points, and votes for the label through majority inside the set.

Different types of data obviously require different values of K, as small alterations to its value can completely change the resulting classification.

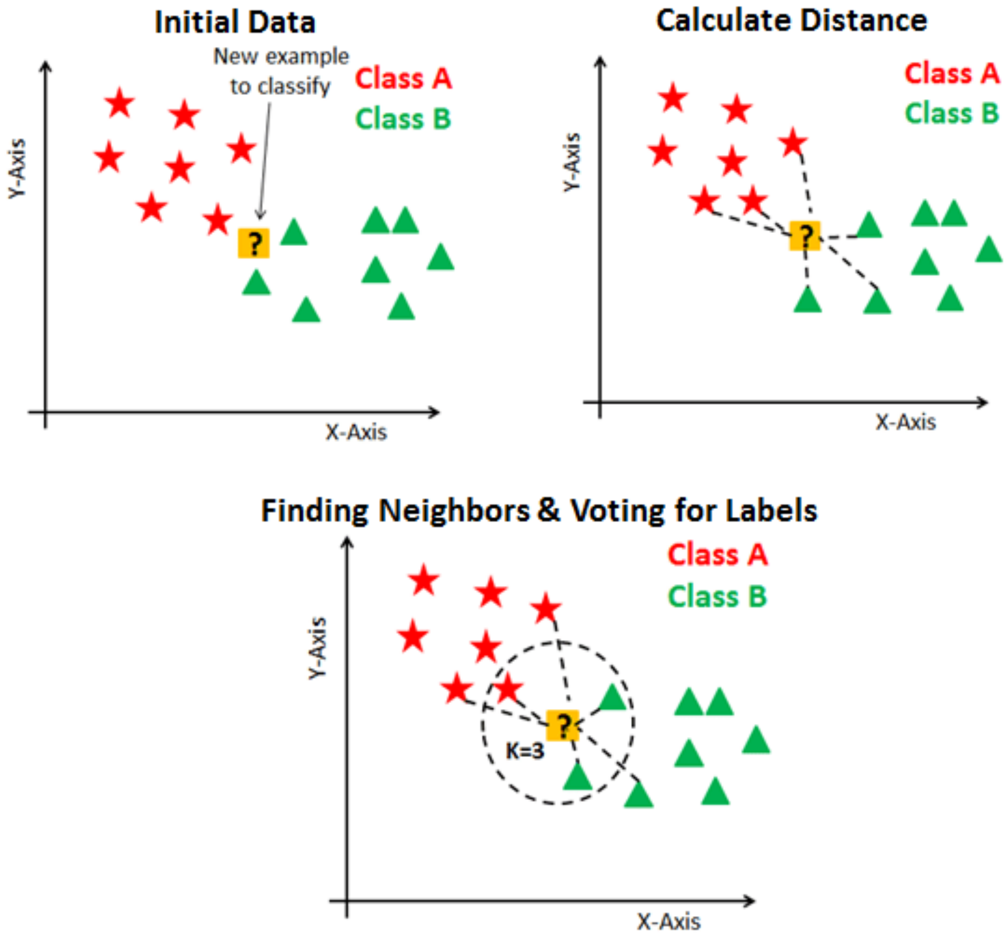


Figure 2.1: K-Nearest Neighbors visualization. Extracted from [22]

2.4.3 Support Vector Machines

Support Vector Machines (SVM) are based on the premise that we find an hypothesis h for which we can achieve the lowest true error possible. This true error is the probability that this current hypothesis fails on correctly classifying a new and random example [23]. SVM work by constructing a hyperplane, or a set of hyperplanes, on the dimensional space, which are then used for classification. The goal is to form a hyperplane whose distance to the nearest point of data is the largest possible, in order to provide a validation error as low as possible. In a text classification problem, SVM performs very well since their ability to learn does not depend on the dimension of the features, meaning that SVM can handle very large feature spaces, as such is the case of text classification. This classifier may use the concept of Soft Margins to solve slight overlap problems. However, when data is strongly not linearly separable, SVM provides different kernels to deal with this problem.

2.4.4 Decision Trees

Unlike the Naïve Bayes, which is a probabilistic approach, Decision Trees use a set of rules to make decisions, categorizing them as rule-based approaches. As a tree like structure indicates, nodes are connected through branches, terminal nodes are called leaves and are situated at the bottom of the tree and the root, which contains all the examples that are to be classified, is at the top.

More specifically, the C4.5 Decision Tree, which is based on the ID3 algorithm [24] and is widely used, is a statistical classifier that works as follows. It initially uses a classified set as input, with each sample in the set consisting of the features as well as the class it belongs to. The algorithm then chooses the feature that split the set into subsets of several classes – the attribute with the highest Information Gain (explained in Subsec. 2.3.5) is the feature upon which the decision is made. It does this recursively until all the data is processed and classified.

A new version of the Decision Tree was created, named C5 Decision Tree, that has many upgrades over the C4.5. It is of faster execution than C4.5, has better memory efficiency, and uses smaller trees whilst achieving the same results, to name a few improvements.

2.4.5 Rocchio Algorithm

The Rocchio algorithm uses a feedback approach, which in sum, is the idea of recursively gathering feedback on queries made on the previous results. Through feedback on those previous queries, we can make decisions about performing a new query, based on the relevancy of the resulting data.

The algorithm represents each document as a vector in a vector space in a way that similar documents have similar vectors, with each space in the vector representing a selected feature. It uses a word weighting heuristic that aims to give more importance to regularly occurring words, while other less regularly occurring words are given less importance [25]. The classifier learns by combining the vectors into prototype vectors, which are created by adding document vectors to all other documents of the class, for every possible class.

To finally classify a new document, it uses the cosine of the prototype vector of each class with the document's vector. After calculating the value of the cosine angle between both vectors, the highest value of the cosine is used to classify the document.

$$H(d') = \underset{c \in C}{argmax}\ cos(\vec{d'}, \vec{c}) \tag{2.12}$$

2.4.6 Neural Networks and deep learning

Neural Networks are structures that are composed of artificial neurons (or nodes), that use a mathematical or computational model for information processing. These neurons are organized in layers, often divided into input, output and hidden layers, with each neuron in a layer usually being connected to all other neurons of the next layer.

Much like the synapses in a biological brain, these neurons transmit signals to the other neurons in subsequent layers which are then processed until an output is produced with the neurons in the output layer.

Each neuron and each connection have weights that either increase or decrease the strength of the signal passed through by neurons, and these are altered as the neural network learns. As such, each neuron has a different influence on the output depending on its weight and bias (or threshold).

The output of these neurons is obtained through the weighted sum of the previous neurons' output, to which a bias is then added. This weighted sum with the bias of the neuron is then passed through an activation function to determine if that particular neuron activates, and feeds the information to neurons in the next layer.

For example, if a neuron N has a bias B_N, and it receives signals from three previous neurons with values n_0, n_1 and n_2, and with w_{N0}, w_{N1} and w_{N2} weighted connections respectively, then the neuron N's output is:

$$\varphi\left[(n_0 \times w_{N0} + n_1 \times w_{N1} + n_2 \times w_{N2}) + B_N\right] \tag{2.13}$$

In the above equation, φ represents the activation function, which may or may not activate depending on the bias value of N, the signal values of n_0, n_1 and n_2, and the type of activation function.

Deep Learning can be seen as a tool based in Artificial Neural Networks [26], which can use many layers of interconnected nodes. It has the ability to provide good accuracy results in supervised classification, but it usually needs substantially large datasets to train.

2.5 Document clustering algorithms and techniques

Clustering is the task of grouping unlabelled data into clusters, a group of data points that are similar to one another based on relations with their surroundings. This can be achieved through different algorithms each with their own way of handling different kinds of data.

Density based clustering algorithms group data based on density in a certain area. The higher the density of data points, the higher the probability of that being assigned as a cluster. Being based on density allows the clusters to form any shape, but lack the ability to assign possible outliers to clusters, thus being ignored.

Centroid based clustering uses centroids in data to form clusters around the data, and each data point is assigned to a cluster based on its distance to the centroid of a cluster.

Hierarchical based clustering (or connectivity based) is usually used on hierarchical data, a type of data that is structured in parent-child relationships in a tree structure (an example would be taxonomy or a file system). The result of the application of the algorithm is a top-down tree of clusters.

Distribution based clustering is based on the probably of a data point being part of a cluster, depending on the distance of the point to said cluster. The higher the distance to the center of the cluster, the lower the probability of it being assigned to it, being inversely proportional.

A possible way to validate clustering consistency is through the Silhouette Method. This method provides a measurement of similarity between a data point and its own cluster, as well as to other clusters. Higher average values of comparison between data points inside a cluster mean appropriate clustering configuration, whereas the opposite means that there are either missing or abounding clusters [27].

2.5.1 Agglomerative Clustering

Agglomerative Clustering is a broad term that shelters several other algorithms and techniques. It is also another name for Hierarchical Clustering which builds nested clusters by successively merging or splitting the data, and is usually represented in a tree-like manner.

The “agglomerative” keyword indicates a “bottom-up” approach, with pairing clusters being merged when moving up the hierarchy, while the contrary (“divisive” clustering) indicates the opposite – instead of merges, splits are done when moving down the hierarchy.

The decision of the action to take is based on a dissimilarity of observed sets, with an appropriate metric and a linkage criterion. The metric can be Euclidean distance, Squared Euclidean Distance and Manhattan Distance for example, while the linkage criteria determines the distance between observations.

2.5.2 BIRCH

Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) is a clustering algorithm devised to handle large quantities of data by incrementally and dynamically clustering them.

Important advantages of BIRCH are that each cluster is made without the scanning of all data, using only the measurements that reflect how close each point is to one another, and it uses the notion that not every data point is relevant for clustering purposes.

It uses Clustering Feature (CF) trees, which are height-balanced trees with a branching factor B and a threshold T - non leaf nodes contain at most B entries, but always satisfying the T threshold, which is the diameter of the branch [28].

The algorithm is divided into 4 phases:

- Phase 1 is the first step in scanning the data and building the CF trees

- Phase 2 is an optional phase that does something akin to phase 1; it scans the leaf entries in the initial trees, to rebuild a smaller one, thus saving memory and grouping together sub-clusters into bigger ones.

- Phase 3 is the clustering phase, which clusters leaf entries together. The result of this phase is a set of clusters that encapsulates the major distribution pattern in the data.

- Phase 4 is also optional, but it is a refinement phase where the algorithm tries to correct inaccurate measurements and refine clusters. It uses the centroids produced in phase 3, and redistributes the data to obtain new sets of clusters.

2.5.3 DBSCAN and HDBSCAN

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a clustering algorithm which groups data points based on density. Clusters are high density areas while outliers are usually in low density areas.

The parameters needed for this algorithm to work are *Epsilon* (ϵ) and a minimum number of points (n_p) which acts as a threshold on the density value. The ϵ is the distance used to locate points near a certain other point.

The algorithm starts by choosing a random starting point, which it considers a potential centroid. Depending on the ϵ value, it counts the neighboring data points to the centroid and compares the number of points inside ϵ range to the threshold value (n_p). An easier way to visualize this is to imagine a circle with ϵ radius from the centroid, and whichever point meets the requirement of being inside the circle, it counts towards the density value. If the value is higher than the *minPoints* threshold, then it is a valid centroid and assign all neighboring points to the same cluster.

Within the newly created cluster, it will sample each point contained inside the ϵ radius circumference, using them to expand the initial cluster through the same procedure as before: find out the neighbors within ϵ range and count them - if the number is bigger than the threshold assign them to the same cluster. It does so iteratively through all neighbors in order to expand the cluster until it can do so no more.

If the cluster can no longer be enlarged, it will repeat the procedure for the remaining non-clustered data points, by picking one randomly and repeating the process previously described [29].

Hierarchical DBSCAN (HDBSCAN) is an extension to the DBSCAN algorithm whose major difference from DBSCAN is, very succinctly, allowing the algorithm to vary the epsilon values, resulting in variable density clusters [30].

2.5.4 Expectation Maximization and Gaussian Mixture Models

The Expectation Maximization (EM) algorithm is used in problems involving two sets of random variables, where one is observable and the other is not, or in other words, it works upon incomplete data by trying to compute the missing values. It consists of estimating the maximum likelihood in two major steps – E-Step and M-Step – in several iterations.

E-step is the expectation estimation steps, where it initially performs the classification of each unlabeled document. The following step is the M-step where it maximizes the likelihood of those estimated values being the true ones.

The ability of the EM algorithm to extrapolate missing data is particularly useful in conjunction with other classifiers and clustering algorithms [4], since it can extrapolate missing values from incompletely classified inputs (a bit like in a semi-supervised approach).

EM is regularly used in order to estimate the parameters for the Gaussian Mixture Model (GMM), which are probabilistic models with the basic assumption that all data was generated from different Gaussian distributions with unknown parameters.

GMM function similarly to the K-Means clustering algorithm (in Subsec.2.5.5) but with key differences: with the help of the EM algorithm, it calculates the probabilities of a document belonging to a cluster through one or more probability distributions, instead of distance from the center. As such, it is able to handle certain shapes of data more efficiently than K-Means, like for example, an oblong shape, because it provides a more statistical approach with variance and standard deviation from the norm, rather than a fixed distance.

It is worth noting that GMM do not produce good results for data distributions that are not Gaussian, making it somewhat hard to use in text clustering, as the features used in this context tend to not be of Gaussian distribution.

2.5.5 K-Means

K-Means falls into the centroid based partitioning clustering techniques, since its main objective is to split data into K partitions, with the constant K being a hyperparameter defined previously relating to the number of total clusters.

Initially, the algorithm may start with a set of documents from the *corpus* and, based on similarity, assigns more documents to these initial representatives. Following this starting approach, a new seed is defined that better indicates the center of the cluster. This seed redefinition is done until the point converges into an unmovable spot and what the algorithm assumes it is the best choice of center for the cluster.

Thus, the main goal of K-Means is to form clusters where the maximum distance between each data point inside the cluster is the minimum possible, whilst maintaining a possible maximal distance between each pair of cluster centroids. The distance may be calculated in several ways, such as Euclidean Distance or Manhattan Distance.

One of the main advantages of K-Means is that it requires very few iterations to converge and can work on very large data sets, but a big drawback is that it is really sensitive to input data and initial centroid creation, as well as higher complexity in similarity calculation if the centroid has a lot of words [31]. Furthermore, if real clusters are not hyper-spherical, K-Means cannot obtain correct clustering.

2.5.6 Spectral Clustering

Spectral Clustering works with graph theory, through graph-based distances between neighboring points. It uses the mathematical notion of the eigenvalues and eigenvectors. The latter is a nonzero vector that, when a linear transformation is applied to it, changes by a scalar factor, while the eigenvalue is the factor by which it is scaled. Besides these mathematical notions, it also uses the Laplacian matrix to represent the graph and search the eigenvalues of the graph's Laplacian matrix in order to find a low dimensional embedding of it.

After calculating the Laplacian matrix and the first eigenvectors, the algorithm uses the first eigenvectors to form another matrix, in which one row defines the features of the graph.

After finding the defining row, it clusters the graph based on those features through another clustering algorithm, like for example K-Means.

2.5.7 Affinity Propagation

The main concept upon which the Affinity Propagation algorithm was based upon, is the concept of passing messages through data points. The clusters are formed through the finding of data points which are representative of potential clusters.

Instead of taking as input the number of estimated clusters in the data set, it uses similarities between data points which indicate how well some specific point is suited to be the “exemplar” (or in other words the representative) for another data point. The number of exemplars found through the computation of the similarity of each data point is the resulting number of clusters.

The algorithm works iteratively by building two matrices - the responsibility matrix r and the availability matrix a. The r matrix quantify how well suited a data point serves as the exemplar for another data point. The a matrix represents how fit a data point is to be selected as the exemplar of another data point.

These matrices are updated throughout the iterations and these iterations stop when either a predefined number of iterations is reached, or when the cluster boundaries remain unchanged [32]. Exemplars whose sum of values for both matrices are positive, are then extracted from the resulting matrix and every point which the exemplar represents is clustered together.

2.6 State of the art

Several researchers have attempted to study and test several techniques in the field of information retrieval, obtaining somewhat interesting results at times and often verifying theoretical results empirically, thus validating the theory behind them.

Within the scope of this dissertation, we are interested in techniques that are proven to work in handling large amounts of text documents in a *corpus* and correctly extracting the necessary features for classification and clustering. We will analyse both supervised and unsupervised approaches in order to help grasp the techniques’ strengths and weaknesses.

Preprocessing The preprocessing techniques used depend heavily on the type of data, the used classifiers/clustering algorithms and the type of feature selection metrics. For example, some authors may prefer stemming [7] over lemmatization [9] in some cases because, despite lemmatization offering better morphological comprehension of processed words, it has a much higher computation complexity than that of stemming. Most of the times, the important part of word preprocessing relies on shrinking the terms down to their “primitive” forms - in an algorithm’s perspective, if we want to know the term frequency of the terms in the document, it makes little difference computing said frequency with the lemmatization or stemmatization of a word.

Feature Selection Yiming Yang and Jan O. Pedersen [18] put to the test five different feature selection methods, these being Document Frequency, Information Gain, Mutual Information, χ^2 statistic and Term Strength, with the help of two classifiers – K-Nearest Neighbors and Linear Least Squares Fit (LLSF).

Regarding results of the metrics used (Precision and Recall), Document Frequency, χ^2 and Information Gain all performed exceptionally well, with Information Gain and χ^2 being elected as most effective in feature selection, followed by Document Frequency, then Term Strength and finally Mutual Information. The authors theorized that the poor performance of Mutual Information was due to favoring rare terms and having a strong sensitivity to probability estimation errors. Document Frequency, Information Gain and χ^2 performances indicate that common terms are informative for these feature selection techniques, since by using a term-removal threshold, up to 98% of unique terms were able to be removed with Information Gain and χ^2, and around 90% for Document Frequency, without losing accuracy.

George Forman presented an in-depth and extensive study on feature selection metrics for text classification [33], following the results from Yang and Pedersen. On this study, Forman measured the performance of each metric in several ways, reaching in the best case, for the Binary Separation metric, values circa 0.7, 0.84 and 0.76 for Recall, Precision and F-measure respectively, in a dataset that was preprocessed with the suffix-stripping algorithm by Porter [7] and a stopword list. Besides the preprocessed dataset, Forman added another dataset of abstracts from computer science papers. In total, they amount to 19 multi-class datasets representing 229 binary classification problems, featuring about an average of 149 classes.

In regard to the metrics previously presented in this dissertation, Information Gain, alongside χ^2, were the metrics that performed better – although Information Gain's performance depended on the number of features used, outperforming χ^2 by a small margin, and being the best overall technique if the validation metric used is solely Precision. Forman also stated that, if one chooses to use two different metrics for cross-validation selection, χ^2 and Information Gain both share a striking correlation in which when one fails to perform correctly, the other may fail as well.

A possible way to extract other features that can be further processed, is to use the n-gram approach to extract important expressions or words from a *corpus*. Joaquim Silva et al. [21] provided an empirical evaluation of different LocalMaxs iterations featuring different n-gram "glue" computation metrics. These metrics were put to the test on a *corpus* with 919254 total words, corresponding to the Lusa - a Portuguese news Agency - news broadcasts. Results indicate that using the *fair dispersion point* normalization provided an increase in MWU extraction precision in all metrics, with the top three performing metrics being SCP_*f* (with "_*f*" denoting fair) with the average precision of 84.90%, followed by ϕ_f^2 with 83.33% and SI_*f* with 81.80% [21]. Another study of the LocalMax algorithm, was conducted on another *corpus*, this being a multilingual European Parliament debate collection. Results further verified SCP_*f* as the elected best metric to use in contiguous (an uninterrupted sequence) MWU extraction with LocalMax. After this election, SCP_*f* was tested on different languages, providing an above 70% precision across 4 different languages - English, French, German and Medieval Portuguese [34]. However, for non-contiguous MWU extraction, Mutual Expectation was tested on a Portuguese Political Debates *corpus* with approximately 300000 words and provided the best results, featuring 90% average precision.

David M. Blei, Andrew Y. Ng and Michael I. Jordan proposed the generative probabilistic model, Latent Dirichlet Allocation (LDA), and tested the model as a dimensionality reduction/feature selection approach in two binary classification experiments on the Reuters-21578 dataset, composed of 8000 documents and 15818 word features [20]. By using a LDA model, with estimated parameters on all the documents on the dataset, and by using a SVM trained on the resulting features reduced by LDA, the authors were able to attain a reduction of 99.6% and still have good classification results. The authors suggested that topic-based representation of the *corpus* may be employed as a fast filtering approach for feature selection.

Classifiers In the text classification field, Yonghong Li and Jain Anil [35] studied the effectiveness of some classifiers. They used the Bag-of-Words feature representation for the documents, meaning that the order of the words does not matter, and with each feature vector representing the words appearing in said document. Stop-words were removed and low frequency words were culled in order to improve the effectiveness of the classifiers. Naïve Bayes, K-Nearest Neighbors, Decision Trees and Subspace Model were tested with Yahoo's news data by extracting the human indexed news in which seven categories were present. Besides testing the models alone, they also combined the classifiers to form an Adaptive Classifier Combination (ACC) but the most important results in this scope, are how the classifiers performed.

Despite all performing reasonably well, Naïve Bayes outperformed Nearest Neighbors and Decision Trees, and Subspace Model on some testing. Furthermore, by reducing the number of classes from seven to five, all classifiers suffered an average of 7% increase in accuracy. Concerning dimensionality reduction, the authors noted that the performance of both Naïve Bayes and Subspace model classifiers rose in accordance to the highest number of features, meaning that they are more accurate the more features there are. Additionally, feature extraction helped improve Decision Tree's accuracy by 4% while not being of any advantage in K-Nearest Neighbors.

Thorsten Joachims [23] explored the usage of SVM in text classification. Joachims theorized that due to the nature of the problem at hand, SVM would be a good classifier since they are able to handle high dimensional inputs and have overfitting protection. Moreover, by assuming most of the features are irrelevant, SVM still performs well by using some of these irrelevant features, when compared to other classifiers. Another point in favor of SVM is that most text categorization problems are linearly separable, and document vectors are sparse - meaning that each document vector has little entries which are not nil. Due to SVM' ability to generalize well in high input spaces, the need for feature selection and dimensionality reduction is lessened.

During the experiments, Joachims compared SVM to other four classfiers: Naïve Bayes, Rocchio Algorithm, K-Nearest Neighbor and C4.5 Decision Tree, tested on two different datasets, with performance being measured with Precision and Recall metrics. Results heavily favored SVM as the best classifier, outperforming the conventional methods on both test sets, regardless of chosen parameters for both versions. This does not mean however that SVM is not prone to parameter sensitivity, as the authors from [3] noted. Out of the four conventional methods that were tested, K-Nearest Neighbors was the best performing classifier on both datasets.

Another study that further proves SVM competence in this domain was conducted by Rikta Sen and Ashish Kumar Mandal [3]. The *corpus* was built by the authors, being composed of 1000 documents with a total of 22218 words, and five possible document categories. After preprocessing the documents by removing stop-words, digits, punctuation and applying the stemming algorithm, the total words in the *corpus* were reduced to 18190. The feature selection metric used was a length normalized $TF-IDF$ weighting vector.

Classifier performance evaluation was again conducted with the Precision, Recall and F-measure metrics, and four classifiers were tested: Naïve Bayes, C4.5 Decision Trees, SVM and K-Nearest Neighbors. Results further prove that SVM outperforms the other 3 classifiers as the average accuracy of the SVM on this *corpus* was 89.14%, followed by Naïve Bayes with 85.22%, then C4.5 Decision Tree with 80.65% and finally K-Nearest Neighbors being the worst with an average accuracy of 74.24%. Other tests were conducted, namely by varying the input to the classifiers by 30 documents between the 5 steps they were trained and tested through. By varying the number of training documents, it is clear that SVM started performing better the more documents there are present in the training set, suffering the highest variation in F-measure score and ultimately having the best accuracy at 150 test documents in the training set. This leads to the assumption that with smaller and more concise training sets, other classifiers perform better, but SVM is still the best classifier in handling large data.

These previously stated approaches were mostly supervised approaches, and to try and contradict that, Bing Liu et al. [4] proposed the usage of the EM algorithm alongside the Naïve Bayes classifier in what the authors called "a semi-supervised approach". The main idea behind this approach is the usage of the EM algorithm to estimate the missing values, since EM can help assign a probabilistic class label in each non-labelled document. Two large *corpora* were used, from which 30 different datasets were created, one with 4 main categories – Computer, Recreation, Science and Talk - the other with 6 categories – Student, Faculty, Course, Project, Staff, Department. Results lead us to believe that using the EM algorithm in conjunction with another classifier would result in extremely accurate results, even if only knowing one positive class of the document.

Bhawna Nigam et al. [36] further proved EM's ability to help create semi-autonomous classifiers, able to estimate missing document labels through previously known ones. The dataset used for this experiment was a "car evaluation" dataset, with 5 features and 3 pre-defined classes, which was split in half in order to create test and training sets. Performance was analysed through the same methods as before, and the main takeaway is that the semi-supervised performed better than the supervised technique. The authors however, note that despite EM being able to effectively estimate and extrapolate data from previously labeled examples, it may not completely translate to other real world scenario as the complexity of most of the text data may not be completely encapsulated in a statistical model.

In an attempt to create a fully unsupervised approach, Youngjoong Ko and Jungyun See [37] proposed a new unsupervised approach, which eliminated the need of manual training document creation. The method consists of creating keyword lists of each category automatically, using “representative sentences” and word and sentence similarity matrices, which are then used to train and classify the documents. In this paper, they focus solely on χ^2 for feature extraction and Naïve Bayes as the classifier. Experimentation was conducted on 47 total categories, with a total of 2286 documents - 1383 for training and 903 for testing. Performance evaluation was computed with the F-measure on both supervised and unsupervised tests, culminating in a 3.8% difference favoring the supervised approach, averaging 75.6%, over the unsupervised approach, averaging 71.8%. Ultimately, this difference between supervised and unsupervised is negligible as the trade-off between performance and *corpus* preprocessing is decent.

Clustering Regarding clustering algorithms, George Seif provided an high-level overview of 4 of the previously mentioned clustering algorithms in [38]. This article glances over the main advantages, disadvantages of the following algorithms: K-Means, DBSCAN, GMM with EM and Agglomerative Hierarchical Clustering, alongside Mean-Shift Clustering. Starting with K-means advantages and disadvantages – K-means offers a linear complexity $\mathcal{O}(n)$, since the gist of the algorithm is distance measuring between points and centroids, culminating in few computations overall. But the main problem is that the results heavily depend on the starting cluster “seeds”, providing inconsistent results, and the input to K-means requires the user to know exactly how many possible clusters there are, which most of the times it is not possible.

DBSCAN offers some advantages, mainly in the form of noise-detection, not requiring a pre-defined number of clusters and working relatively well with arbitrarily sized and shaped clusters. A major flaw however, is that due to the way the algorithm works with fixed ϵ and a minimum number of points threshold, it may not find variable density clusters. This flaw is remedied in the HDBSCAN extension of the algorithm, allowing the algorithm to find variable density clusters.

EM and GMM provides a bigger level of flexibility when handling different types of data. As previously mentioned in Subsec. 2.5.4, EM and GMM are able to handle ellipsoid shapes due to the way it handles covariance and standard deviation of the cluster. By being a probabilistic model it also supports mixed-membership, meaning a point can effectively belong to two or more classes. A huge drawback to GMM, is that for any non-gaussian distributed dataset, they perform very poorly.

Finally, Agglomerative Hierarchical Clustering does not require the user to specify the number of clusters like K-Means, being advantageous in some cases, and it is somewhat universal when it comes to distance metric selection, contrary to most other clustering algorithms which are metric sensitive. The main drawback is the enormous time complexity, being that of $\mathcal{O}(n^3)$.

For more information regarding comparisons between clustering algorithms and their benchmark performances and scaling on datasets, HDBSCAN Documentation [39] provides a comparison of ten different clustering algorithms on different increasingly larger datasets, randomly generated with Numpy Python library. Alongside this performance comparison, the Scikit-learn, a Python machine learning library, documentation also features visual comparison of how clustering algorithms clustered different types of datasets and in-depth explanation of the algorithms and their parameters in the implementation [40].

Now regarding published papers with algorithm comparisons in the text and document clustering field, in [41], the authors compared K-Means, Spectral Clustering and Affinity Propagation. They began by tokenizing and stemming the text and weighing the terms with $TF-IDF$, followed by a similarity matrix using cosine similarity. The supervised *corpus* was composed of 60 problems, with each problem containing 20 texts, and algorithm performance was evaluated with F-measure, Recall and Precision. Post-experimentation, the authors noted that Affinity Propagation had the best averaging Precision, followed by Spectral Clustering then K-means, with 0.704, 0.694, 0.619 respectively. With the Recall measure, the order changed to Spectral Clustering being the best performing, then K-means, then Affinity propagation, with 0.833, 0.747, 0.606 respectively. Finally, with the F-measure metric, the order stayed the same as the Recall metric with the following values: 0.758, 0.677, 0.651 for Spectral Clustering, K-means, and Affinity Propagation.

The authors theorized that Spectral Clustering performed better than the other two algorithms because of the dataset in question; as Spectral Clustering works better with few clusters, while Affinity Propagation works better with a large number of clusters. K-means was the worst of the three, possibly because it is randomly initialized, that is, before it tries to converge to a local optimum it randomly assigns a centroid as a "seed" for a cluster to grow around, and this initial randomness may heavily impact the final results. It is worth noting again that Affinity Propagation, besides the previously mentioned attribute of working better with large amounts of data, does not take as input the number of clusters *a priori*, meaning that Affinity Propagation is a good algorithm to use when the number of possible clusters is not known at the start.

Focusing on cluster topic identification, Michael Snow [10] empirically tested the usage of Singular Value Decomposition (SVD) to extract possible cluster topics. SVD is a linear algebra technique to decompose a matrix into three other matrices, with which we are able to create the best low rank approximation of the initial matrix. Michael Snow initially processed the text with removal of words that do not contribute to the learning of the system, alongside punctuation and digits. Post-processing, documents were vectorized with Doc2Vec in order to create a vector space in which every document is embedded in. Clustering was done with HDBSCAN, post t-Distributed Stochastic Neighbourhood Embedding dimensionality reduction [42], which provided a two dimensional representation of the Doc2Vec vector space upon which the clusters were built. Results on nearly 87000 documents comprising of paragraphs describing businesses showed more than 4000 identified separate clusters through HDBSCAN, which makes manual cluster labelling near impossible. By assuming that each document in a cluster is highly similar to one another, SVD can be applied to each of the document vectors for each cluster. This results in a rank 1 representation of the matrix which is then compared with cosine distance to all other documents' matrices in the original vector space, with the closest vector being returned. This document can then be used as the descriptor of the cluster as a whole, or can be used to manually infer the topic. The authors state that in a very noisy case, SVD may not be applicable as it may chose a noisy vector as the representative of that cluster.

2.7 Chapter conclusion

The text mining field is a very vast and technologically dense field, with a plethora of possible usable algorithms and techniques, each with their own pros and cons. The choice of these techniques heavily depends on the type of problem at hand as well as the final objective, the approach – supervised or unsupervised – and also language dependency. This, in turn, makes it quite difficult to accurately say which of these techniques is the best overall. However, there are some empirical studies that prove that some algorithms may be more suitable than others in the text-mining domain.

Previously, the most used and most known techniques have been laid out and explained at a very high level in hopes of providing the necessary knowledge to understand more complex and newer techniques in the field, often built upon the foundations of these techniques, as well as the proposed solution that will be explained in the following section.

3

Proposed approach for unsupervised clustering and classification of documents

The following chapter goes into further detail about the proposed approach and our contribution to the problem at hand.

Unsupervised approaches to clustering and classification problems face a big challenge as, since the documents classes are not known, we can not train the system the same way we would in a supervised approach. To do so, it is imperative to extract the best possible features as to correctly identify the documents classes.

3.1 Feature selection

Since there are no class labels, the assessment of the quality of each *candidate feature* is not easy to obtain. Nevertheless, there are some metrics that may indicate how informative a *candidate feature* is. As we are dealing with unstructured text, words with stronger semantics tend to discriminate the topic of the document – take for example the average length of the *candidate feature* words and the $TF\text{–}IDF$ (Subsec. 2.3.2) value of the singular words. In fact, the *candidate feature* "biological conservation", having an average length of 11 characters, is semantically stronger than "shoe's sole", whose average length is 4.5; And it is expected that "biological conservation" discriminates the documents of a *corpus* much more than "shoe's sole".

Firstly, we need to select the best candidate words and expressions (*candidate features*) in the text documents to further apply those metrics. This step was done with the application of the LocalMaxs algorithm (Sec. 2.9) to extract relevant expressions of size between two and seven. Do note that it is not possible to apply LocalMaxs to singular words, as for there to be glue between words there needs to be a minimum of two words composing a relevant expression. As such, distinct unigrams were all accounted for and further refined, as succinctly explained in the next subsection.

3.1.1 Stop-word removal

Most of the singular words comprising the documents hold little to no semantic meaning, that is, they provide nothing of value in regards to understanding a document's class. Usually, these words appear quite commonly throughout the texts and can be discarded with barely any information loss.

As such, stop-words were removed through a stop-word array of the Natural Language Toolkit (NLTK) [43] Python library, by selecting only the words that do not appear on this stop-word array.

Previously extracted n-grams were also refined through the usage of the NLTK stop-word array by purging all n-grams (with $n \geq 2$) that started with, or ended with, any of the stop-words appearing in the stop-word array or any special characters.

The usage of the NLTK library stemmed from the need of having readily available stop-words at our disposal, as using another approach – such as the "elbow method" – required a bigger *corpus* in order to correctly identify the stop-words.

It is important to note that, by using NLTK's stop-word array, we are not adopting a language independent approach, since the stop-word array only accounts for words in the English language. The usage of the "elbow method" or any other statistical approach would ensure language independence and, as such, would be preferable in a *production* setting.

Throughout the following sections, and unless stated otherwise, the usage of the expression "n-gram" also encapsulates unigrams, "relevant expressions" include both n-grams and unigrams, and "term(s)" is used interchangeably with "relevant expression(s)".

3.2 Feature dimensionality reduction through Similarity Matrices

By using the relevant expressions extracted by the LocalMaxs (Subsec. 2.3.8) algorithm and in order to reduce the number of obtained features, a similarity matrix between documents was built in such a way that the quality of features should ensure that similarity values are high between pairs of documents that we know to be of the same class, and low between documents of different classes.

These similarities were computed through the Pearson Correlation Coefficient, that will have into account the discriminating power, that is, the quality of each feature. Note that each document is initially characterized by the number of occurrences of each feature already then weighed by its quality. Thus, each document will be characterized by the similarity it has with all other documents of the entire *corpus*, forming a new set of features equal in size to the number of documents, which represents a stronger reduction as the initial number of attributes was composed of all extracted and filtered relevant expressions.

Covariance measures how the variation of a variable is related to the variation of another. In the context we are dealing with, two documents d_i and d_j can be taken as two variables. It is measured as followed, with $|T|$ denoting the cardinality of the set of n-grams:

$$\begin{gathered} cov(d_i, d_j) = \frac{1}{|T|} \sum_{t \in T} (V(t, d_i) - V(\cdot, d_i)) \times (V(t, d_j) - V(\cdot, d_j)) \\ \text{and} \\ V(t, d_n) = P(t, d_n) \times Q(t) \end{gathered} \tag{3.1}$$

Where $Q(t)$ can be any combination of the metrics explained in the following subsections. Through results shown in Ch. 4, it is visible that the usage of any singular metric was not enough to obtain the needed quality of the similarity matrices, such that good results are achieved in clustering. This imposes some experiments on the usage of several combinations of metrics.

By interpreting this formula (Equation 3.1), it is clear that if a certain term t appears in both documents, it is a positive influence on the covariance, and it is also a positive influence if it does not appear in either document. The influence of a term t is only negative if its occurrence is only in one of the documents in the expression.

And finally, the Pearson Coefficient reflects the correlation between two documents:

$$S(d_i, d_j) = \frac{cov(d_i, d_j)}{\sqrt{cov(d_i, d_i)} \times \sqrt{cov(d_j, d_j)}} \tag{3.2}$$

$S(d_i, d_j)$ is one of the $N \times N$ cells of the similarity matrix. The resulting matrix has N rows and N columns, where N is the number of documents that were used to compute the matrix. Values inside the matrix are within the range of $[-1, 1]$, meaning positive values signal that the documents share some features/expressions. The higher the positive value and the closer to 1, the higher the similarity between the documents and thus the desired result for documents of the same class.

Obviously, in a *production* scenario, classes will be unknown as these must be suggested by the approach. However, during the development of this dissertation's proposed approach, we needed to evaluate the quality of the obtained results. To do so, we need to work with human supervision, which is provided by the classes that we know each document belongs to. Let us call these *meta-classes*.

3.2.1 Variation Coefficient

The Variation Coefficient tries to measure how disperse the probability of each term t is. It is computed by the standard deviation of the probability of term t through the documents, divided by the average probability of t in the same documents. As such, the equation for the Variation Coefficient is:

$$Cv(t) = \frac{\delta_{P(t)}}{\mu_{P(t)}} \tag{3.3}$$

With t being the n-gram term whose Variation Coefficient we are calculating, $P(t,\cdot)$ in Equation (3.4) is the average probability of the n-gram's appearance in a document of the *corpus*, and $|Docs|$ is the number of documents in the *corpus*.

$$Cv(t) = \frac{\sqrt{\frac{1}{|Docs|}\sum_{d_i \in Docs}(P(t,d_i) - P(t,\cdot))^2}}{\frac{1}{|Docs|}\sum_{d_i \in Docs} P(t,d_i)} \tag{3.4}$$

3.2.2 Skewness Coefficient

The main idea behind the usage of the Skewness Coefficient measurement is to try and understand which terms show to be outliers in the distributions of their probability through the documents. Thus, if a term t has a significantly higher probability in a small set of documents than in others, then the Skewness Coefficient value is positive, suggesting that t is characteristic of that small set. If the Skewness Coefficient of t is close to zero, it means that t does not characterize that set of documents. Cases of negative Skewness Coefficient of t are usually close to zero, meaning that t is also irrelevant. From now on, in this dissertation document, we can write “Skewness” or “3rd Moment” to reference the Skewness Coefficient.

Besides the 3rd Moment, which is usually the statistical moment used in the Skewness metric, we also attempted to calculate the 5th Moment of Skewness, alongside Kurtosis (4th Moment), in order to empirically test if there are any improvements in the document similarity matrices obtained through them.

We define a general metric $Sk(t,n)$ in Equation (3.5), with t being the n-gram on which we want to calculate the coefficient of Moment n, $|Docs|$ being the number of documents in the *corpus*, and $P(t,\cdot)$ is the average probability of the n-gram's appearance in a document of the *corpus*. When $n = 3$, we are calculating the Skewness; If $n = 4$ the Kurtosis will be returned, etc..

$$Sk(t,n) = max\left\langle 0, \frac{\frac{1}{|Docs|}\sum_{d_i \in Docs}(P(t,d_i) - P(t,\cdot))^n}{\frac{1}{|Docs|}\sum_{d_i \in Docs}\left[(P(t,d_i) - P(t,\cdot))^2\right]^{\frac{n}{2}}}\right\rangle \tag{3.5}$$

3.2.3 Probability Jump

The Probability Jump metric – or Jump for short – aims to detect significant changes in the word's probability inside the documents of the *corpus*. To do so, for every term t, we order the documents the term appears in by descending order of probabilities. The reasoning behind this ordering is that we intend to single out terms that only appear with higher probabilities in certain classes, as these tend to better discriminate the class of the document. The formula is as such, with t being the n-gram we want to compute the

Jump of, $\overline{P}(t)$ is the average probability of t in the *corpus*, and $\Delta_i(t)$ is the difference in probability of t in documents d_i and d_{i+1}:

$$
\begin{gathered}
Jump(t) = \frac{1}{(n-1)\times\overline{P}(t)} \times \sum_{i=1}^{i=n-1} \Delta_i^2(t)\times F_i \\
\text{and} \\
\Delta_i(t) = P(t,d_i) - P(t,d_{i+1}), \qquad P(t,d_i) \geq P(t,d_{i+1}) \;\; \forall\, i
\end{gathered}
\tag{3.6}
$$

To further illustrate how this metric works, take, for example, two terms t_i and t_j and the ordered probabilities of said terms in ten documents, five for each class:

$$P(t_i) = [0.400,\ 0.380,\ 0.370,\ 0.370,\ 0.365,\ 0.360,\ 0.350,\ 0.340,\ 0.340,\ 0.310]$$

$$P(t_j) = [0.200,\ 0.190,\ 0.185,\ 0.183,\ 0.180,\ 0.020,\ 0.018,\ 0.015,\ 0.014,\ 0.014]$$

By applying this metric to these two terms, t_j would be more valued in comparison to t_i, as it appears more often in a select few group of documents, and we can assume that it is a term characteristic of that class, contrary to t_i as it has very little variation in probability in all ten documents, signaling it as very common in both classes. In order to further increase the significance of the probability difference between documents, $\Delta(t)$ is squared and is multiplied by F_i, which is the attenuation factor being measured as such:

$$
F_i = \begin{cases} (\frac{2i}{n})^{\frac{1}{2}} & , i \leq \frac{n}{2} \\ \\ (2-\frac{2i}{n})^2 & , i > \frac{n}{2} \end{cases}
\tag{3.7}
$$

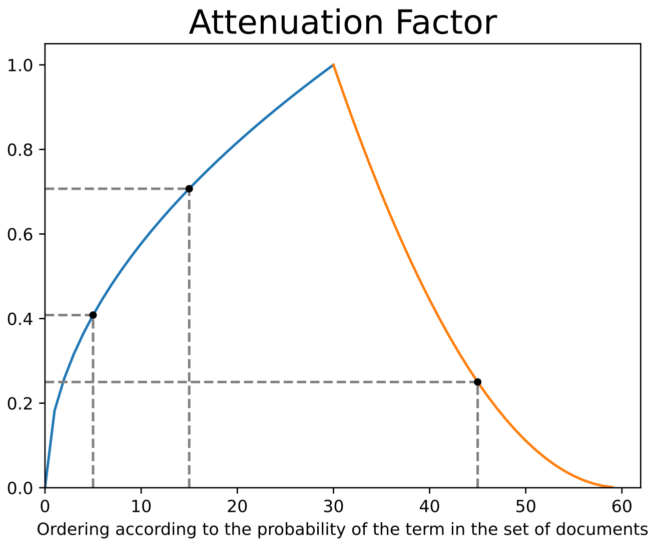


Figure 3.1: Graphical example of Attenuation Factor (F_i in Equation (3.6) and Equation (3.7)) with n = 60

This factor aims to provide higher weight to terms whose biggest jumps in probability occurs close to $\frac{n}{2}$, considering the ordering of the n documents of the *corpus*, according to the probability of t. In other words, the later the biggest jump appears in the ordered probabilities, the higher it will be valued. In turn, if the jump occurs after the $\frac{n}{2}$ position, we start devaluing the difference in probabilities as the term is too common. This can be visualized more easily in Figure 3.1, where it's clear to see that F_i gives a higher weight to the first and second markers, but gives a lower weight to the third marker, as it appears in a lot more documents and the function starts devaluing after the $\frac{n}{2}$ mark (in this case, n is equal to 60). The curve in Figure 3.1 favours terms where a high jump in probability occurs "before" $\frac{n}{2}$, in comparison to those where the jump occurs after that point. For example, if a term t occurs in $\frac{1}{3}$ of the documents of a *corpus*, it is more likely that it is characteristic of some class(es) of documents, rather than if t occurs in $\frac{2}{3}$ of the documents.

3.2.4 Additions to the metrics

Besides having used the previous metrics in the $V(t, d_n)$ expression in the covariance computation (Equation (3.1)) of the Pearson Correlation Coefficient (Equation (3.2)), and in the hopes of increasing the resulting similarities, the previous metrics were multiplied by several other factors used alternatively, that we deemed as plausibly good additions. This in turn changed the previous expression to $V(t, d_n) = P(t, d_n) \times Q(t) \times Addition$ with $Addition$ being one of the following factors in Subsubsec. 3.2.4.1, 3.2.4.2, and 3.2.4.3, and Subsec. 3.2.5.

3.2.4.1 Average term length

$$\overline{L}(w_1, \ldots, w_n) = \frac{1}{n} \times \sum_{i=1}^{i=n} len(w_i) \tag{3.8}$$

Average term (or word) length is one way to give more weight to longer, more discriminating expressions. Take for example the following expressions: "Biological Agriculture" (average 10.5), "Football Championship" (average 10), "Politics" (average 8), "Albeit"(average 6); It is clear that those expressions whose average word length is higher, are more discriminative of the document topic. This leads to the assumption that, in general, discriminating expressions tend to be longer than non-discriminating ones, even regarding unigrams.

3.2.4.2 Median

The median aims to achieve the same as the average term length, that is, value expressions that posses a higher amount of characters. However it offers better results in certain cases, mainly where the expression has small words - such as definite articles like "a"/"an" – which weigh the average down. For example "International Astronomical Union" has an average term length of 10, while the median is 12. Thus, by ignoring the lengths that are smaller, it tends to favors the longer words, which are more semantically meaningful.

3.2.4.3 Expression size

There is no doubt that, generally speaking, expressions consisting of two or more terms are more discriminative than others. The idea behind the usage of the size of an expression is to give more weight to expressions such as “International Union of Chemistry” or “Milky Way Galaxy” as opposed to singular terms that commonly hold little to no discriminative power.

3.2.5 *W* function

The *W* function aims to give more weight to terms based on a few following characteristics: the popularity of a term, the distinct number of *n*-grams of the same size in the document, and the average length of the words in the relevant expression.

Term popularity is defined by being the logarithm of the number of document this term appears in. Much like the F_i equations explained beforehand (Equation (3.7)), we intend to give higher value to expressions that appear in several documents at the same time. ϵ is an infinitesimal value, as to ensure that very common terms are extremely undervalued when compared to not so common terms. With t being a term we wish to compute the popularity of, and n the number of documents t appears in, we get:

$$Popularity(t) = \begin{cases} \log(n+1) & , 1 \leq n \leq \frac{|Docs|}{2} \\ \\ \epsilon & , n > \frac{|Docs|}{2} \end{cases} \tag{3.9}$$

To give further importance to the terms, we use the number of distinct terms in a document to boost the value of the metric. For all relevant expressions of size equal to the relevant expression we want to compute the *W* function for, we count the total distinct relevant expressions in said document. It is worth noting that “size” refers to the number of words in the relevant expression – so for example “Chemical Properties” has size two. Let t be a term and d the document we want to compute the total number of terms of size equal to t:

$$D(t,d) = |t' \in d \wedge t' \neq t \wedge size(t') = size(t)| \tag{3.10}$$

These two previously explained notions, along with the average length of the expression (Subsubsec. 3.2.4.1), culminate in the following expression which is then used in conjunction with the metrics in the covariance computation (Sec. 3.2):

$$W(t,d) = Popularity(t) \times D(t,d) \times \overline{L}(t) \tag{3.11}$$

Ultimately, only the Skewness (3rd Moment) metric was tested with the *W* function, as it was our elected metric to compute the similarity matrices, due to attaining the best results in general, by comparison with Variation Coefficient (Subsec. 3.2.1), Probability Jump (Subsec. 3.2.3) and other Skewness Moments.

3.3 Feature reduction through PCA

Despite being a major improvement over the tens of thousands of features (relevant expressions) extracted by the LocalMaxs algorithm, the computation of the similarity matrix still yielded a big feature space that needed further reducing.

Principal Component Analysis is a widely used and highly effective method of reducing features that works by summarizing how each feature relates through one another through their covariance.

Through the computation of the covariance matrix between the features of a multidimensional feature set, and after finding the eigenvectors and eigenvalues of said matrix, PCA is then able to select the N best principal components that capture most of the cumulative variance of the dataset.

The number of principal components vary depending on the use case, and there is no standardized way of selecting the optimal number of principal components. One possible way to determine the optimal number of components is to see the cumulative explained variance ratio and select the one which has the best trade-off of dimensionality reduction and variance retention - in other words, the higher the number of components, the higher the cumulative variance of the features, but the higher the number of retained features.

For our specific case, we use two principal components as they provided better results overall, and easier visualization of the resulting data.

3.4 Clustering

After the application of the PCA to our similarity matrix, it is now possible to cluster the documents into singular categories. This was done by giving the results of the PCA transform to several clustering algorithms – BIRCH (Subsec. 2.5.2), Spectral Clustering (Subsec. 2.5.6) and GMM (Subsec. 2.5.4) – whose results will be displayed in Ch. 4.

One problem that arose was the necessity of explicitly stating the number of clusters for the algorithms to function correctly. It is worth noting that, as our approach was developed and tested, documents needed to be labeled for human confirmation of the resulting clusters, and we had to know *a priori* the number of clusters – which isn't possible in unsupervised problems. But there is a way to know what is the optimal number of clusters, which is based on the Silhouette Method.

The Silhouette Method measures how similar a data point is to its own cluster when compared to other clusters. Resulting values range from -1 to +1 and it reflects how well the point was clustered. The average of all points of the dataset validate how well the clusters were built, with values closer to 1 being a good result. The method is defined as such, with $a(i)$ being the mean distance between sample i and all other points in same cluster, and $b(i)$ the mean distance between the sample i and all other points of the nearest different cluster:

$$s(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}} \tag{3.12}$$

After testing the algorithms with several iterations of the number of clusters parameter, and obtaining the Silhouette Coefficients of said iterations, the best coefficient is chosen and the appropriate number of clusters is obtained.

3.5 Classification of new documents

Post-clustering, we want to classify new documents based on what the system has learned through the previous phases. New entries to the system are raw text, which means that for these entries to be classified as one of the learnt classes, we need to transform the text the same way we previously did, with a few key differences.

Classification cross-validation was done with Leave-One-Out technique that consists in using all of the instances of data in the dataset with the exception of one, which is later used in testing. So for example, in a *corpus* of 60 documents, we would get 60 folds of cross validation with the first one being *Test: [0]* and *Train: [1,2...,59]*, then *Test: [1]* and *Train: [0,2...,59]* and so forth. Since we are testing with smaller datasets it is possible to apply this cross-validation technique, as it can be very costly for big datasets due to the amount of folds equaling the number of samples.

For the sets of training documents, we need to compute the relevant expressions and their corresponding SCP_*f* glue values through the LocalMaxs algorithm (Subsec. 2.3.8) – if they're composed of two or more terms – and metrics (stated in Sec. 3.2), as these obviously change depending on the documents used in training.

After the extraction of the relevant expressions we then proceed to the computation of the similarity matrix using the Pearson Correlation Coefficient (Equation 3.2) for the $N - 1$ training documents in the fold. PCA is used to reduce the number of features, and is fit with the resulting training documents' PCA scores, which will be used to transform the test document matrix entry into the feature space of the training documents.

But before the test document is transformed by PCA and subsequently classified, it needs to be compared with those existing in the training set. As such, a new matrix entry is computed that consists of the similarity between the test document and those in the training similarity matrix – this will result in an entry of $(N - 1)$ size, consisting of the test document's similarity to those in the training set.

These similarities will be transformed by the PCA into the feature space of the training data and classified by SVM (Subsec. 2.4.3), which was given the predicted labels of the training document group and the PCA score of the test document entry to classify.

It is also worth noting that all the words and expressions that consist in the new entry are not taken into probability and metric calculations. This means that only words appearing already in the system and in the new entry are used for computations.

3.6 Extracting the content of the clusters

In order to understand the given topic of a cluster, as the clustering algorithm just labels the clusters numerically, the most important expressions were extracted from the documents inside the clusters. This provides the user with some insight on what the given cluster topic is about, which is very important for human readability and result validation.

In order for an expression to be elected as a topic discriminant, it needs to be ranked among other relevant expressions found in documents of the same cluster.

In the case of being an n-gram consisting of two or more words, we used the SCP_*f* glue of the LocalMaxs algorithm (Subsec. 2.3.8) since higher glue values usually mean that those expressions have high semantic meaning and, as such, are good candidates as discriminants of a cluster's topic. For n-grams of length two and higher we compute the quality initially computed as such:

$$Q_n(RE, C_i) = SCP\_f(RE) \times \frac{|d \in Docs(C_i) \wedge f(RE, d) > 0|}{|d \in Docs(C_i)|} \times G(RE) \tag{3.13}$$

The second factor of Equation (3.13) stands for the ratio of documents that contain the relevant expression. The factor $G(RE)$ will be fulfilled with combinations of Skewness and Median metrics. Results will be shown in Sec. 4.5.1.

However, some variations regarding the addition of other metrics and word length were taken into consideration and will be shown in Ch. 4.

For unigrams, the equation above needs to be different as firstly, they do not have the SCP_*f* glue as it is purely intrinsic to the LocalMaxs algorithm (Subsec. 2.3.8) and requires at least two or more words to compute the glue, and secondly, unigrams tend to be more common in several other categories instead of just one. To counteract this, an adaptation of $TF{-}IDF$ (Subsec. 2.3.2) was used, as this variation shows how important a term is in relation to a cluster:

$$TF{-}IDF_{cluster}(t, d) = P(t, d) \times \log\left(\frac{|Docs|}{|d_i \in Docs(C_i) \wedge f(t, d_i > 0)|}\right) \tag{3.14}$$

Which is used in the following equation:

$$Q_u(t, C_i) = \frac{1}{|Docs(C_i)|} \sum_{d \in Docs(C_i)} TF{-}IDF_{cluster}(t, d) \times G(t) \tag{3.15}$$

The factor $G(t)$ in Equation (3.15) will be fulfilled with the inclusion, or not, of the length of the single word. Results will be shown in Sec. 4.5.2.

After the computation of these values for all the relevant expressions in the cluster's documents, we order them in descending order and choose ten in total. We extracted five n-grams and five unigrams, as we deemed them sufficient in understanding the cluster's topic. Do note that this choice was somewhat subjective and arbitrary, as we could've chosen more or less expressions. Still, we needed a good equilibrium on readability and understanding of a cluster's topic – for example having only two expressions may be insufficient in understanding the topic, whilst having twenty expressions would help understand the topic better but have unnecessary and redundant information.

3.7 Proof of concept - Indirect Expressions

At times, certain documents of the same *meta-class* do not share many expressions. This can happen quite often in very broad topics such as "Politics" and "Philosophy", that often have highly different subtopics within their scope.

To counteract this, we propose a new concept in this domain – the usage of *Indirect Expressions*. Initially, pairs of expressions are computed in such a way that every unique term that is directly adjacent to any other term is accounted for. Afterwards, it is just a matter of finding pairs of expressions in the likes of $t_a \leftrightarrow t_b \leftrightarrow t_c$, with t_b being the expression that links t_a and t_c, and both t_a and t_c do not appear in the same document.

To fully understand Indirect Expressions, we need to understand the notion that a direct expression is one that directly contributes to the similarity of a document with another document. That is, an explicit expression of both documents.

For example, if a document d_i has the expression "Atom", then "Atom" is an explicit expression of document d_i. If we now take into consideration documents d_i and d_j, and if d_j has the explicit expression "Hydrogen", it is possible to define "Hydrogen" as an indirect (or implicit) expression of d_i. This can occur if and only if, there is an expression that links both "Atom" and "Hydrogen", that exists in both documents (d_i and d_j), and is adjacent to both "Atom" and "Hydrogen". Do note that the usage of the word "expression" encompasses both unigrams and n-grams up to seven words in size.

For a visual representation, let t_n be the expression existing in both documents, that links both expressions "Atom" and "Hydrogen" and admit the following correlation $S(.,.)$ values:

$$Atom \xleftrightarrow{S(Atom,t_n)=0.5} t_n \xleftrightarrow{S(t_n,Hydrogen)=0.8} Hydrogen$$

$S(.,.)$ is computed through the Pearson Correlation Coefficient, as such:

$$S(t_i,t_j) = \frac{cov(t_i,t_j)}{\sqrt{cov(t_i,t_i)} \times \sqrt{cov(t_j,t_j)}} \tag{3.16}$$

$$cov(t_i,t_j) = \frac{1}{|Docs|} \sum_{d \in Docs} (P(t_i,d) - P(t_i,\cdot)) \times (P(t_j,d) - P(t_j,\cdot))$$

This correlation defines how similar an expression is to one another with regards to their common documents. Higher values reflect that the expressions appear almost exclusively in the same documents, while lower values mean that one of the expressions appear in more documents than the other.

Furthermore, the indirect correlation between "Atom" and "Hydrogen" will be equal to $S(Atom,t_n) \times S(t_n,Hydrogen)$ that, with a high enough threshold, should be able to extract indirect correlations that allow us to boost the similarity between documents of the same *meta-class*. For this, we used two thresholds – the *first threshold* was used to cull common pairs of expressions and is applied when the pairs are computed, which allows for swifter computations of the pairs that really matter. The *second threshold* is applied when computing the similarity matrices, as this allows us to have an alterable threshold for testing, and observe how changes to the *second threshold* value influence the similarity matrices.

However, if we are to take Indirect Expressions into account, we also have to artificially enhance the length of a document, as we cannot simply use an expression that never appeared in a document in the computation of the similarity between two documents, subsequently providing incorrect results due to discrepancies in the probability of expressions.

To artificially enhance the length of a document, we sum all the maximum correlations of expression not found in the document, with their indirect counterparts found in the document. More formally, let S_{t^*} be the set of Indirect Expressions of a document d, and t_i^* be a Indirect Expression such that:

$$Size^*(d) = Size(d) + \sum_{t_i^* \in S_{t^*}} \max_k(S(t_i^*,t_k)), \forall t_k \in d \tag{3.17}$$

Through this formula we can compute a "*pseudo-size*" for any document with some significance given to Indirect Expressions. This will be crucial in probability computation since, when computing the similarities between documents, we need to use the new *pseudo-size* of the document they belong to as it needs to be reflected on the probability of an expression to avoid erroneous results. As such, for all explicit expressions in any document, we get:

$$P(t,d) = \frac{f_{t,d}}{Size^*(d)}, \forall t \in d \tag{3.18}$$

And for every implicit expression with a correlation to any term in d:

$$P(t_i^*, d) = \frac{\max_k(S(t_i^*, t_k)) \wedge t_k \in d}{Size^*(d)}, \forall t_i^* \notin d \wedge t_i^* \in S_{t^*} \tag{3.19}$$

Finally, these correlations need to satisfy a threshold as to not be overwhelmingly abundant in the computations of the similarities between documents, and to help separate indirect terms from other *meta-classes*. This threshold should be high enough as to select some indirect terms of the same *meta-class* since, if this threshold is too loose, it may start taking into account implicit terms of another *meta-class*, resulting in an increase of similarity between documents of two highly different *meta-classes*.

By using only the maximum correlation of the implicit expression towards any explicit expression in d, we can ensure that no implicit expression will be overvalued when compared to explicit expressions. This is because the correlation values will always be between zero or one and, as such, any correlation that satisfies the threshold condition will always be maximized by one, which is the equivalent of the lowest possible frequency of any expression in d.

3.8 Chapter summary

From raw text to similarity matrices, the text needs to be heavily processed in order for the system to fully function.

The process starts with extracting the unigrams and n-grams – called relevant expressions – through LocalMax (Subsec. 2.3.8) and further refinement by removing stop-words. Post extraction, these relevant expressions are used to compute the similarity matrices with several metrics, and the resulting similarity matrices are then built with Pearson Correlation Coefficient (Sec. 3.2).

These similarity matrices then get further reduced by PCA, and are clustered by clustering algorithms (Sec. 3.3 and Sec. 3.4). These clusters are validated with the Silhouette Method, which also helps us define the best numbers of clusters to input as a parameter to the clustering algorithm.

For new document entries to be classified, these need to be compared with documents already in the system. To do so, words in the new entry are compared to previously existing relevant expressions in the system, and the resulting set of common words are used for the similarity computations. This step results in a similarity array that holds the similarities of the new document to all others existing in the system at the time.

After the computation of the similarity array of the new/test document, it is then transformed by the PCA algorithm to the number of components previously chosen and it is posteriorly predicted by a trained classifier (Sec. 3.5).

In order to understand the main topic of each cluster, we used two approaches to extract the main expressions; one for unigrams and another approach for n-grams. For n-grams, we took advantage of the LocalMaxs' glue value since, generally, highly discriminative words have higher glue value between them. For unigrams, we adapted $TF\!-\!IDF$

to work with clusters, by using the frequency of the unigram in the whole cluster instead of just a single document.

Finally, in this chapter we introduced our Proof of Concept regarding the usage of Indirect Expressions. These were planned to aid in the computation of similarities between documents of the same *meta-class*, that feature a low common expression count. Indirect Expressions aim to exploit the transitive relations between expressions found in several documents, which in turn allow some correlations to be made between said documents, further boosting their similarity obtained through the Pearson Correlation Coefficient.

Since the correlation between any two Indirect Expressions is never higher than the smallest frequency of any word in a document, it will never be overvalued when compared to explicit expressions of the documents. Furthermore, the size of the documents is altered in order to adapt to the new amount of expressions that are now being added to the similarity computation. This is done through the sum of all maximum correlations of all indirect terms that feature a high enough correlation with any explicit expression in a document.

Presented in Figure 3.2 is a pipeline diagram that shows the steps executed by our approach in order to process the text from a given *corpus*.

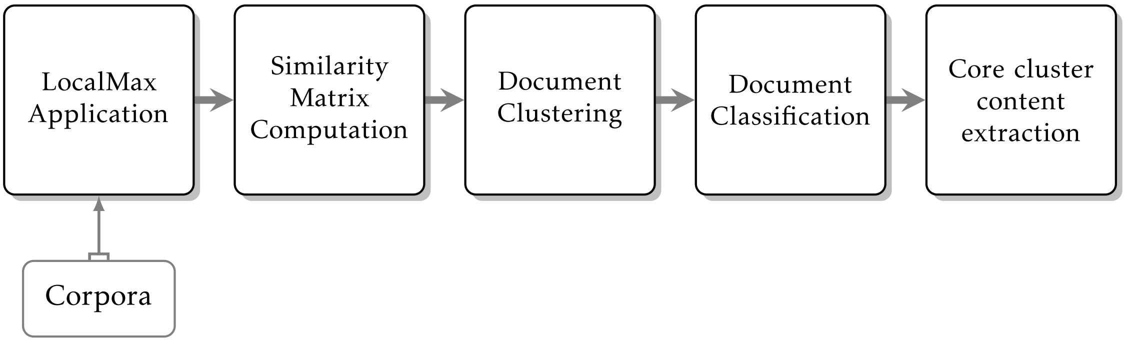


Figure 3.2: Pipeline diagram of our approach's phases.

4 Results

This chapter will present and compare the results obtained by our solution in all phases, obtained with two different corpus. Corpus I consists of 58 documents of three different categories and corpus II consists of 107 documents of four different categories.

Initially, to validate our approach, we attempted to test a *corpus* with three highly different categories – Constellations, Tennis Championship Finals and Chemicals – extracted from certain categories from Wikipedia, that shall be named *corpus* I. This way, by having three different categories with barely anything in common (with the exception of constellations and chemicals, that share a very small subset of words) we can guarantee our approach is working with very high performance. Following the initial validation of our approach with *corpus* I, we tested with another, more *difficult corpus*, composed of four categories that fall under the main-category of "Animals" – these categories are Cats, Dogs (more specifically, Hounds), Birds and Fish – for a total of 107 documents. The documents present in *corpus* II, much like *corpus* I, were extracted from Wikipedia articles.

4.1 LocalMaxs and Relevant Expression extraction

The application of the LocalMaxs algorithm on *corpus* I yielded decent results. For all the 58 documents in the *corpus* and a total of 12588 total words – culminating in 65223 possible distinct combinations of expressions of size two through seven – LocalMaxs was able to extract 486 n-grams between said sizes, which is almost a 96% reduction in size when compared to the number of total words. Through further refinement (Subsec. 3.1.1), we were able to achieve 276 total extracted n-grams.

Similarly, when applied to *corpus* II, with a total of 55258 words – resulting in 287143 distinct combinations of expressions – LocalMaxs extracted 2403 n-grams. This initial reduction (approximately 99%) in size was, through further refinement, reduced even further to a final value of 1397 n-grams.

These final subsets of 276 and 1397 n-grams extracted from *corpus* I and *corpus* II respectively, are the expressions that will be used in the similarity computations.

Overall accuracy of the LocalMaxs algorithm was good, providing n-grams such as "International Astronomical Union", "2nd-century Astronomr Ptolomey", "Wimbledon Championships" and "Applied Chemistry", for *corpus* I, and "breed of domestic cat", "large hunting dog" and "Red-breasted flycatcher" for *corpus* II.

Regarding unigrams, the total number of distinct unigrams in *corpus* I is 2648 which, upon further reduction through the usage of the NLTK library and punctuation removal, were lowered to 2253. Further reduction in unigrams, through the same means as *corpus* I but applied to *corpus* II, yielded a reduction from 7234 initial unigrams to 6568.

4.2 Similarity matrices

The overall best result was obtained with Skewness (Subsec. 3.2.2) multiplied by Jump (Subsec. 3.2.3) and the W function (Subsec. 3.2.5), or $PSkJW$ for short, as will be shown later. This statement is true for both *corpora*.

4.2.1 Variation Coefficient matrices

As we can see in Appendix A, the resulting matrices computed with the Variation Coefficient used to fulfill component $Q(t)$ of Equation (3.1), Sec. 3.2, are quite lackluster in quality, as there are too few decent correlations between documents in order to cluster them correctly. Through testing, we came to the conclusion that the Variation Coefficient suffers a major drawback as, for the documents to have high similarity between them, they require a high number of equal relevant expressions which is very rarely the case. In figure A.3 (with documents from *corpus* I) there are some relatively high similarity values regarding this subsection, as all of these documents are about the "Wimbledon Championship Finals" that took place throughout the years and, as such, possess highly similar expressions and terms like "Championship tennis match" or "Open Men's singles final".

When tested with the average word length, expression size and expression median, the resulting matrices improved slightly, although not nearly enough to fully cluster the documents together, as some similarities are still low, as depicted in Figure A.4, which refers to a section of the matrix obtained with the variance in conjunction with the median.

For *corpus* II, the same is true, meaning that Variation Coefficient also performed poorly in providing good enough similarity matrices in order to efficiently cluster the documents. For the sake of brevity, and due to the nature of *corpus* I and the obtained results with the Variation Coefficient and other metrics (aside from $PSkJW$), resulting similarity matrices from metrics applied to *corpus* II will be omitted, as they would be redundant. The reasoning is that since *corpus* I had such lackluster results with some metrics, and *corpus* II is a significantly more *difficult corpus* to work with, it would result in even worse matrix quality.

4.2.2 Skewness (3rd Moment) matrices

Resulting similarity matrices computed with the Skewness (Subsec. 3.2.2) metric to fulfill $Q(t)$ in Equation (3.1), yielded overall better results than with the Variation Coefficient, but the similarity matrices still had pretty low average value between documents of the same class as Figures B.1, B.2 and B.3 portray.

Slight variations on the matrix occur when the Skewness is used in conjunction with the three previously mentioned additions (Subsec. 3.2.4). In certain cases the similarity would rise by a small margin, in other cases the similarity would lower slightly, ultimately resulting with about the same average similarity per document as just the Skewness. Figure B.4 shows the best section of the matrix obtained with the Skewness in conjunction with the Median metric (Subsubsec. 3.2.4.2) when applied to files from *corpus* I, and it is clear that despite these small increases it is still not enough to correctly cluster the documents.

However, major differences occur when the W function (Subsec. 3.2.5) was used in conjunction with the Skewness and, despite the major upgrade over previous iterations of the matrices, this result was still not sufficiently good.

It was through further testing with the Jump (Subsec. 3.2.3) metric in conjunction with Skewness and W function that we were able to achieve the best matrices. These new matrices – depicted in Figures C.1, C.2 and C.3 for *corpus* I, and Figures C.4, C.5, C.6, C.7 for *corpus* II – possess high similarity values between documents of the same *meta-class*, and low similarity values between documents of different *meta-classes*, which allows us to proceed to the next phase since having good similarity matrices is imperative for good clustering and classification of documents.

From henceforth every mention of matrices in the clustering and the classification phase, will refer to matrices that were computed with the Skewness, Jump, and the W function with $V(t,d_n)$ (from Equation 3.1) as such:

$$V(t,d_n) = PSkJW(t,d_n) = P(t,d_n) \times Sk(t,3) \times Jump(t) \times W(t,d_n) \qquad (4.1)$$

4.2.3 Matrices with 4th and 5th Moments

Contrary to the improvements of the Skewness matrix over the Variation Coefficient matrix, the Kurtosis (4th statistical Moment), when used to fulfill $Q(t)$ in Equation (3.1), was a direct decline in matrix quality, as the similarities were overall lower. We believe that the exponent increase between the Skewness and Kurtosis reflects a bigger sensitivity of the metrics towards data, which resulted in lower similarities between documents as we can see in Figures D.1, D.2 and D.3.

Without much surprise, the increase in exponent value yielded worse results yet again, as the 5th Moment (Equation 3.5) was the worst of the three (3rd, 4th, and 5th Moments), by the same reason previously stated. This can be viewed in Figures D.4, D.5 and D.6.

4.2.4 Jump matrices

Computed matrices using the Jump metric (Subsec. 3.2.3) to fulfill $Q(t)$ in Equation (3.1), were overall pretty lackluster in quality, however, not nearly as much as 4th and 5th Moments. The Jump metric was able to offer higher values in some cases when compared to the previously mentioned metrics.

However, the overall matrix computed with the Jump metric is not nearly enough to fully cluster the documents correctly since, as Figure E.1 details, similarities between documents of the same *meta-class* are worse than Skewness (3rd moment). However, there are some documents that feature high similarities such as the pair (45,53) found in Figure E.3, but a few really high similarities are not enough for good clustering results, and for *corpus* II the same is true.

4.3 Clustering results

Through the usage of a mixture of metrics and other additions, we were able to obtain a very good similarity matrix. As mentioned previously, the matrix that we used to cluster the documents together was computed with $PSkJW$ (Probability, Skewness, Jump and W function), but, despite this being our elected metric of choice, we also tested the clustering with Skewness (isolated) to see the resulting clusters, which are displayed in Figure F.1 and Figure F.2 for documents in *corpus* I, and Figure F.5 and Figure F.6 for documents in *corpus* II. Figure F.1 and Figure F.5 depict Skewness (isolated) metric results, and Figure F.2 and Figure F.6 depict $PSkJW$ metric results of their corresponding *corpora*.

We tested the clustering with three different algorithms, these being Spectral Clustering (Subsec. 2.5.6), BIRCH (Subsec. 2.5.2) and GMM (Subsec. 2.5.4). On top of each of the subplots in the figures of Appendix F there is the average Silhouette Score of each of the combinations of algorithms with variable number of clusters. It is clear that the best results were obtained with a number of clusters equal to three in all cases, with all algorithms, for *corpus* I, and for *corpus* II the best result was obtained with the number of clusters equal to four.

Comparing both approaches' Silhouette Score, the $PSkJW$ metric scored higher values when compared to the isolated Skewness metric. As expected in *corpus* I, with a similarity matrix of high quality, the clusters are quite separated and easily distinguishable when obtained with the $PSkJW$ metric, but with the usage of the isolated Skewness metric the clusters are closer to one another resulting in a lower Silhouette Score (Figure F.1).

Figure F.5 and Figure F.6 show the comparison between the clustering of both isolated Skewness and $PSkJW$ metric, respectively, when applied to documents of *corpus* II. It is clear that the $PSkJW$ metric offers a better result regarding how the documents are clustered and, through the Silhouette Score of each subplot, we can also assert that the best parameter for the number of clusters is four, as mentioned previously. Similarly to what occurred in *corpus* I, the usage of the isolated Skewness metric resulted in a convergence of clusters towards the center point, ultimately causing overlapping clusters and an overall lower Silhouette Score in this *corpus*.

These comparisons also allows us to easily identify the best number of clusters to use as a parameter for the clustering algorithm, as ideally we would want as high of a Silhouette Score as possible, since this reflects that the clusters are well separated from each other and each cluster has correctly assigned data points.

It is worth noting again that, since we need validation on how well the algorithm clustered the documents together, we inherently know the *meta-classes* that these documents belong to, which is not possible in a *production* setting. The Silhouette Method allows our approach to easily discern the correct number of clusters to input as parameter for the algorithm, which in this case is three.

Regarding algorithm choice, Spectral Clustering performed slightly worse in *corpus* II, but it was chosen for posterior testing as it inherently works with similarity matrices, and all three algorithms have somewhat equivalent results when applied to both *corpora*.

4.3.1 Clustering Precision and Recall values

Regarding the resulting clusters obtained with the Spectral Clustering (Subsec. 2.5.6) with our metric of choice ($PSkJW$), and through the confusion matrix presented in the table below, we can compute the Precision and Recall values associated to the clustering of documents from *corpus* I.

		Predicted class		
		Chemicals	Constellations	Tennis
Actual class	Chemicals	16	0	0
	Constellations	0	22	0
	Tennis	0	0	20

Table 4.1: Confusion matrix of Spectral Clustering results for documents from *corpus* I

$$Precision_{Chemicals} = \frac{16}{16} = 1 \qquad Recall_{Chemicals} = \frac{16}{16} = 1$$

$$Precision_{Constellations} = \frac{22}{22} = 1 \qquad Recall_{Constellations} = \frac{22}{22} = 1$$

$$Precision_{Tennis} = \frac{20}{20} = 1 \qquad Recall_{Tennis} = \frac{20}{20} = 1$$

$$Precision_{Global} = \frac{1+1+1}{3} = 1 \qquad Recall_{Global} = \frac{1+1+1}{3} = 1$$

Indeed, the separated and somewhat dense cluster shapes in Figure F.3 suggests the high Precision and Recall values of the clustering process.

By comparison, *corpus* II does not exhibit such good results as *corpus* I, since the documents composing *corpus* II fall under the umbrella term of "Animals breeds" and possess some similarities between documents of dissimilar categories, while also displaying a lower average similarity between intracategory documents. Regarding document distributions throughout the *corpus*, there are 25 documents about "Birds", 32 about "Cats", 22 about "Fish" and 28 about "Dogs", and a visualization of *corpus* II can be seen in Figure F.4. Through this image, we are able to see the results of the clustering obtained by Spectral Clustering. Precision and Recall were computed taking into account the real distribution of documents through the correct clusters and the predictions.

		Predicted class			
		Birds	Cats	Fish	Dogs
Actual class	Birds	25	0	0	0
	Cats	0	32	0	0
	Fish	4	0	18	0
	Dogs	0	2	0	26

Table 4.2: Confusion matrix of Spectral Clustering results for documents from *corpus* II

$$Precision_{Birds} = \frac{25}{25+4} = 0.86 \qquad Recall_{Birds} = \frac{25}{25} = 1$$

$$Precision_{Cats} = \frac{32}{32+2} = 0.94 \qquad Recall_{Cats} = \frac{32}{32} = 1$$

$$Precision_{Fish} = \frac{18}{18} = 1 \qquad Recall_{Fish} = \frac{18}{18+4} = 0.82$$

$$Precision_{Dogs} = \frac{26}{26} = 1 \qquad Recall_{Dogs} = \frac{26}{26+2} = 0.93$$

$$Precision_{Global} = \frac{0.86+0.94+1+1}{4} = 0,95 \quad Recall_{Global} = \frac{1+1+0.82+0.93}{4} = 0.9375$$

4.4 Classification results

Classification validation was conducted through "Leave-One-Out Cross Validation", which needed matrix re-computation for each of the training sets of documents (as previously mentioned in Sec. 3.5).

An example of the computed similarities for the new document – "Tennis" file of index 50 in *corpus* – can be visualized in Table G.1 and it is very clear that there is an enormous gap in difference between similarities of documents in the same *meta-class* – "Tennis" in this case – and documents of different *meta-classes* existing in *corpus* I.

This enormous gap in similarity allows the classifier to more accurately pinpoint the class of the test document, and also allows the PCA transform to further project the test document towards its true cluster.

Figure G.1 shows how the previously mentioned document – whose similarities are displayed in Table G.1 – would be clustered together with its respective cluster after the similarity array was computed and transformed by PCA.

Throughout the various folds of the cross validation applied to *corpus* I, SVM always correctly predicted the class of the test document, which amounts to a Precision and Recall value of 1.

For *corpus* II, a few images are shown in Appendix G that illustrate some classifications done by SVM (Figure G.2 and Figure G.3). It is worth noting that Figure G.3 showcases a document that was wrongly classified, as it originally belongs to the "Fish" *meta-class* but was misclassified as being of the "Birds" *meta-class*. Furthermore, these figures differ from those shown previously for *corpus* I, as they also possess an extra plot for the Silhouette values of the points of each cluster. This plot helps visualize the Silhouette value of all the points of all the obtained clusters, which is helpful in determining whether we achieved a good clustering result or not. It is to be noted that this method requires human verification and validation of the plots, and it is mostly for demonstrative purposes in this dissertation, since what we use for the unsupervised approach is the computed average Silhouette Score.

		Predicted class			
		Birds	Cats	Fish	Dogs
Actual class	Birds	25	0	0	0
	Cats	0	32	0	0
	Fish	6	0	16	0
	Dogs	0	2	0	26

Table 4.3: Confusion matrix of SVM classification results for documents from *corpus* II

$$Precision_{Birds} = \frac{25}{25+6} = 0.81 \qquad Recall_{Birds} = \frac{25}{25} = 1$$

$$Precision_{Cats} = \frac{32}{32+2} = 0.94 \qquad Recall_{Cats} = \frac{32}{32} = 1$$

$$Precision_{Fish} = \frac{16}{16} = 1 \qquad Recall_{Fish} = \frac{16}{16+6} = 0.73$$

$$Precision_{Dogs} = \frac{26}{26} = 1 \qquad Recall_{Dogs} = \frac{26}{26+2} = 0.93$$

$$Precision_{Global} = \frac{0.81+0.94+1+1}{4} = 0.9375 \qquad Recall_{Global} = \frac{1+1+0.73+0.93}{4} = 0.915$$

As expected, Precision and Recall values are lower due to certain documents being overlapped with documents from a cluster that is not of the same *meta-class*, such as the document demonstrated in Figure G.3.

4.5 Cluster topic extraction results

Regarding cluster identification, we extracted ten expressions through the means explained in Sec. 3.6 (Equations (3.13) and (3.15)) and they are overall of good quality, which in turn allow us to quite easily discern the topic of a cluster.

Some variations to the Equations (3.13) and (3.15) were also tested, as to further refine the expressions extracted from the clusters. These variations are explained in the following Subsections.

4.5.1 Extracted expressions with size two or greater

With the initial equation of SCP_*f* times the ratio of documents that have a certain relevant expression in the cluster (Equation (3.13)), and with $G(RE) = 1$, we obtained the first group of expressions. The extracted expressions from *corpus* I can be found in Table H.1. Some are of decent quality, but there are other expressions that have little to no discriminative power or are too ambiguous for us to understand the topic of – like for example "*Earth's crust*" and "*88 modern*".

We intended to counteract this through fulfilling $G(RE)$ in Equation (3.13) to $Median(RE)$ metric or fulfilling $G(RE)$ to $Sk(RE,3) \times Median(RE)$ (Sk in Sec. 3.2.2). Both of these approaches' extracted expressions can be seen in Table H.2 and Table H.3, respectively.

Regarding Table H.2 in comparison to Table H.3, there is no clear best choice as both have equally decent terms that encapsulate the clusters' topic well.

Extracted expressions from *corpus* II are shown in Table H.6, Table H.7 and Table H.8 for $G(RE) = 1$, $G(RE) = Median(RE)$ and $G(RE) = Sk(RE,3) \times Median(RE)$, respectively.

Overall, extracted expressions aren't great, but are decent enough to help understand the topic of the cluster in most cases. The most glaring cases are the "Fish" and "Birds" clusters, as they are composed of expressions that are quite ambiguous at times, and offer little to no discriminative meaning – such as "Union for Conservation of Nature" and "Canary Islands". This can be counteracted by increasing the number of extracted expressions by a small margin, in order to cull out ambiguities and help solidify the understanding of the cluster's topic.

4.5.2 Extracted expressions with a singular term

Akin to the expressions with size two or greater, we also extracted five expressions composed of a singular term. The extracted expressions from *corpus* I can be found in Table H.4, which was obtained with Equation (3.15) with $G(t)$ resolving to $Length(t)$, and in Table H.5, obtained with Equation (3.15) with $G(t)$ resolving to 1. These results offer very good insight on the clusters' topic, with only one at best per *meta-class* being of poor quality.

This allows us to further explain why five is a decent choice for number of extracted expressions. As with five terms, even if we get a few outliers and worse expressions, we can easily understand the topic of a cluster.

Extracted expressions from *corpus* I were of good quality but there are a few expressions that offer no insight regarding the *meta-class* they belong to – for example "2014", "Open" and "degrees" – and as such we opted for Equation (3.15) with $G(t) = Length(t)$ as our one-word topic extractor.

Results from *corpus* II are decent in both tests – shown in Table H.10 for $G(t) = 1$ and Table H.9 for $G(t) = Length(t)$ – with some slightly better expressions being obtained with $G(t) = 1$.

4.5.3 Quality of extracted expressions

Concerning the quality of the set of extracted keywords (unigrams) and key terms (n-grams) extracted from the clusters to inform about the core content of the clusters, we can take the Precision and Recall metrics. Regarding unigrams extracted from *corpus* I, Table H.4 shows 0.8 Precision for the "Constellations" cluster, as only "represents" is not informative whilst "Orion", "Centaurus", "Triangulum" and "astronomical" are content revealing. For "Chemicals" and "Tennis" clusters we can see that Precision is 1 and 0.8 respectively. Additionally, for *corpus* II and by the using the same $G(t)$ fulfilment in Equation (3.15) as used for *corpus* I (resulting in the expressions presented in Table H.9), we get a Precision value of 0.4, 1, 1, 1 for the "Birds", "Cats", "Fish", "Dogs" clusters respectively. It is worth mentioning that the scientific name of the species present in the extracted expressions were deemed correct for the computation of the Precision metric, as long as it from a species that belong to the cluster it is extracted from (for example, "*Gymnocephalus*" is a genus of ray-finned fishes but it is in the "Birds" *meta-class* cluster).

For the n-grams case, Table H.2 shows that "element with the symbol" is not perfect, whilst "chemical element with the symbol", "atomic number", "periodic table" and "alpha particles" clearly show the topic content of the cluster, corresponding to 0.8 precision in the "Chemicals" cluster. For "Constellation" and "Tennis", Table H.2 allows us to compute 1 and 0.8 respectively. By using the same $G(RE) = Median(RE)$ fulfilment in Equation (3.13) as used for *corpus* I, we get the following Precision values for the "Birds", "Cats", "Fish", "Dogs" clusters that compose *corpus* II, whose values are 0.2, 0.6, 0.4, 0.4 respectively (extracted from expressions in Table H.7). Resulting Precision values are as good as if we were to use expressions from Table H.6, as otherwise these would have been 0.2, 0.8, 0.6, 0.4 for the same *meta-classes* previously mentioned.

However, due to lack of time, it was not possible to assess the Recall since, firstly, it would be necessary to check, based on some manual procedure, the top 5 most informative n-grams and the top 5 most informative unigrams in each cluster. This would be a requirement to compute Recall.

4.6 Results obtained with Indirect Expressions

Similarity matrices computed with our approach regarding Indirect Expressions (Sec. 3.7) were quite satisfactory. Initially, we extract pairs of expressions with a *first threshold* of 0.25 (Equation (3.16)). This first filter assures that only pairs of terms appearing in a significant number of documents are considered for Indirect Expressions.

The similarity matrices for *corpus* I were computed with four *second threshold* values (Sec. 3.7), these being $0.75, 0.7, 0.65, 0.60$. Performance was best when the *second threshold* was above 0.70, as it was a strict enough threshold to allow Indirect Expressions to influence the result, but not loose enough to allow the usage of an abundance of Indirect Expressions that would inflate the similarity between documents of different *meta-classes*. For brevity, only select parts of the matrices will be displayed in Appendix I.

In comparison with the metric used in our best resulting matrix (Equation (4.1)), the addition of Indirect Expressions resulted in a great increase in the similarity of some documents, as can be seen in Figure I.1, in which the overall worst subsection of any computed matrix previously is shown – documents regarding "Chemicals".

This approach has shortcoming, as briefly mentioned in Sec. 3.7, since Indirect Expressions may link two documents of completely distinct *meta-classes*, which is obviously not desirable (Figure I.2 show an example of the undesired effect of Indirect Expressions in documents of different *meta-classes*). This can be mitigated with a highly precise threshold (*second threshold*).

Clustering and classification results with Indirect Expressions were quite decent, as we can see in Figure I.3 and I.4. The Silhouette Score is higher as the clusters are somewhat more dense than without the usage of Indirect Expressions, and the test document is clustered very closely to the centroid of the cluster it belongs to.

Since *corpus* II has more documents than *corpus* I, and these documents are generally of bigger length, the thresholds used by the Indirect Expressions needs to be further tuned in order to reduce the possibility of an increase in similarity between documents of different *meta-classes*. The *first threshold* used in the computation of pairs of expressions was increased to 0.5, in contrast to *corpus* I 0.25 *first threshold* mentioned previously, as to eliminate even more unwanted pairs of expressions. The main reasoning here is that, since the total number of extracted expressions in *corpus* II is much higher than *corpus* I, the higher *first threshold* would help further reduce the computation time needed to find the pairs of expressions that are more important.

Furthermore, the *second threshold* is much more precise, due to the amount of available Indirect Expressions after the initial culling still being quite high. For illustrative purposes, Figure I.6, Figure I.7 and Figure I.8 showcase how a 0.01 change in the *second threshold* alters the structure of the clusters.

We want to maximize the Silhouette Score obtained by the clustering and, as such, the *second threshold* had to be fine-tuned to maximize the Score. For this *corpus*, we found that a *second threshold* value of 0.9045 provided a slight increase in Silhouette Score compared to the $PSkJW$ metric without Indirect Expressions. Figure I.5 shows the original clustering with $PSkJW$ metric only, and while we can see that Figure I.8 averages the same Silhouette Score, the 0.9045 *second threshold* slightly increases this Silhouette Score to 0.609 – as Figure I.9 shows.

5 Conclusions

The following chapter will give some final considerations about our approach, what was proposed and what was accomplished, and some possible improvements for future work.

In the domain of unsupervised clustering and classification of data, the lack of known labels is an added challenge that needs to be overcome. The first challenge is feature selection, as features need to be judiciously selected in order for the clustering and classification phases to output good results.

Feature selection was done through the LocalMaxs (Subsec. 2.3.8) algorithm that, despite sometimes extracting sub-par expressions (interchangeably called *n*-grams), is language independent and the resulting sub-set of expressions is a direct reduction of features, as otherwise we would have to use most, if not all, of the *corpus*' words.

Despite being a direct reduction of size, the resulting expressions were still far too plentiful and needed further reduction. Through the creation of similarity matrices with the Pearson Correlation Coefficient (Sec. 4.2), we were able to further reduce the number of features from thousands or tens of thousands, to only the number of documents in the *corpus*. These matrices reflect the similarities between documents, with documents of the same *meta-class* featuring a high similarity between them, and documents of different *meta-classes* featuring the opposite.

The resulting matrices were then further reduced through PCA and used in three clustering algorithms, all of which had very similar results between them regarding our initial number of *meta-classes*.

The clustering and classification phase required our approach to re-compute all of the relevant expressions and similarity matrices for each Leave-One-Out cross-validation folds, so that a new similarity array of the test document could be compared to documents in the training set. In a *production* setting, a new entry to the system would also require a similar workflow to be converted and posteriorly classified. Results with SVM proved quite satisfactory as most of the predictions done in the cross-validation were correct, meaning that the features extracted from the documents are of good enough quality.

Topic extraction is also important in our approach, as it provides insight on what the topic of the documents in the cluster is. Overall results proved to be helpful in understanding the topic of obtained clusters.

When cluster and classification results are high in unsupervised context, an advantage stands out: the dynamic set of possible classes is automatically mined by the approach. In this dissertation , we tried to contribute on this domain.

5.1 Final considerations

At the beginning of this dissertation, we proposed a system that should be able to:

- Extract good enough features from a *corpus* to allow the approach to function in a fully unsupervised manner.
- Cluster documents according to their categories, through similarity between documents.
- Correctly classify new documents based on what was learned previously, and cluster them accordingly.
- Extract the core content of a cluster through topics.
- Keep language independence.

These goals were all met, and the results proved to be quite positive. Furthermore, we introduced a new concept – Indirect Expressions (Sec. 3.7) – that aims to capitalize on the transitive relations between expressions of different documents, allowing us to mitigate the problem of same class documents with few common expressions.

It is important to note that, since LocalMaxs is an algorithm that is able to extract features from a *corpus* regardless the language of written text found in it, our system is fully language independent since, as mentioned previously, the usage of a previously built stop-word array is due to *corpus*' size constraints in statistical approaches.

5.2 Future work

Despite achieving good results, there are still some optimizations that can be done to improve our approach.

Firstly, any optimization conducted in the feature selection and extraction phase would result in a direct increase in the following phases' quality. As such, improvements to set of relevant expressions extracted by the LocalMaxs algorithm would directly translate into higher quality similarity matrices and posterior clustering and classification. One possible improvement that can be done, is to further refine the set of extracted expressions by removing relevant expressions that may be redundant. Another possible optimization would be to give weights to the components that are used by the metrics. This could possibly result in better similarity matrices, by fine-tuning the weights in order to better emphasize certain aspects of the relevant expression. This optimization, should it result in better matrices, would directly improve the clustering and classification phases.

Lastly, regarding Indirect Expressions, a possible optimization would be to give more weight to certain expressions based on their SCP_*f* in order to favor terms that are highly discriminative of a document's class, as well as use only a few Indirect Expressions per document in order to try and mitigate the problem shown at the end of Ch. 4.

The results show that the improvements obtained from the Indirect Expression concept is promising but needs further investigation.

Bibliography

[1] J. M. Lourenço. *The NOVAthesis LaTeX Template User's Manual*. NOVA University Lisbon. 2021. URL: https://github.com/joaomlourenco/novathesis/raw/master/template.pdf (cit. on p. ii).

[2] F. Sebastiani. "Machine Learning in Automated Text Categorization". In: *ACM Computing Surveys* 34.1 (Mar. 2002), pp. 1–47. ISSN: 0360-0300. DOI: 10.1145/505282.505283. URL: https://doi.org/10.1145/505282.505283 (cit. on p. 1).

[3] A. Kumar Mandal and R. Sen. "Supervised Learning Methods for Bangla Web Document Categorization". In: *International Journal of Artificial Intelligence & Applications* 5 (Oct. 2014). DOI: 10.5121/ijaia.2014.5508 (cit. on pp. 2, 4, 5, 21).

[4] B. Liu et al. "Partially Supervised Classification of Text Documents". In: *International Conference on Machine Learning*. 485. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002, pp. 387–394. ISBN: 1558608737 (cit. on pp. 2, 16, 21).

[5] D. Reforgiato Recupero. "A new unsupervised method for document clustering by using WordNet lexical and conceptual relations". In: *Inf. Retrieval* 10.6 (Oct. 2007), pp. 563–579. DOI: 10.1007/s10791-007-9035-7 (cit. on pp. 4, 6).

[6] A. Doucet and M. Lehtonen. "Unsupervised Classification of Text-Centric XML Document Collections". In: *International Workshop of the Initiative for the Evaluation of XML Retrieval*. Vol. 4518. Springer. Dec. 2006, pp. 497–509. ISBN: 978-3-540-73887-9. DOI: 10.1007/978-3-540-73888-6_46 (cit. on p. 4).

[7] M. Porter. "An algorithm for suffix stripping". In: *Program: electronic library and information systems* 14 (July 2006), pp. 130–137. DOI: 10.1108/00330330610681286 (cit. on pp. 4, 18, 19).

[8] A. G. Jivani. "A Comparative Study of Stemming Algorithms". In: *Int. J. Comp. Tech. Appl.* 2.6 (Nov. 2011), pp. 1930–1938 (cit. on p. 5).

[9] V. Balakrishnan and L.-Y. Ethel. “Stemming and Lemmatization: A Comparison of Retrieval Performances”. In: *Lecture Notes on Software Engineering* 2 (Jan. 2014), pp. 262–267. DOI: 10.7763/LNSE.2014.V2.134 (cit. on pp. 5, 18).

[10] M. Snow. *Unsupervised Document Clustering with Cluster Topic Identification*. Tech. rep. Office for National Statistics, Jan. 2018 (cit. on pp. 5, 23).

[11] C. Ding and X. He. “*K*-Means Clustering via Principal Component Analysis”. In: *Proceedings of the Twenty-First International Conference on Machine Learning*. ICML ’04. Banff, Alberta, Canada: Association for Computing Machinery, July 2004, p. 29. ISBN: 1581138385. DOI: 10.1145/1015330.1015408. URL: https://doi.org/10.1145/1015330.1015408 (cit. on p. 6).

[12] G. Miller. “WordNet: A Lexical Database for English”. In: *Communications of the ACM* 38.11 (1995), pp. 39–41. DOI: 10.1145/219717.219748 (cit. on p. 6).

[13] T. Liu et al. “An Evaluation on Feature Selection for Text Clustering”. In: *Proceedings of the 20th international conference on machine learning (ICML-03)*. 2003, pp. 488–495 (cit. on p. 7).

[14] W. Wilbur and K. Sirotkin. “The automatic identification of stop words”. In: *Journal of Information Science* 18.1 (Feb. 1992), pp. 45–55. DOI: 10.1177/016555159201800106 (cit. on p. 7).

[15] T. Mikolov et al. “Efficient Estimation of Word Representations in Vector Space”. In: *Proceedings of Workshop at ICLR* 2013 (Jan. 2013) (cit. on p. 7).

[16] T. Mikolov et al. “Distributed Representations of Words and Phrases and their Compositionality”. In: *Advances in Neural Information Processing Systems* 26 (2013), pp. 3111–3119 (cit. on p. 7).

[17] Q. Le and T. Mikolov. “Distributed Representations of Sentences and Documents”. In: *31st International Conference on Machine Learning, ICML 2014* 4 (May 2014), pp. 1188–1196 (cit. on p. 7).

[18] Y. Yang and J. Pedersen. “A Comparative Study on Feature Selection in Text Categorization”. In: *Proceedings of the Fourteenth International Conference on Machine Learning*. Vol. 97. ICML ’97. June 1997, pp. 412–420 (cit. on pp. 8, 11, 18).

[19] L. Galavotti et al. “Feature Selection and Negative Evidence in Automated Text Categorization”. In: *Proceedings of KDD*. July 2001 (cit. on p. 8).

[20] D. M. Blei, A. Y. Ng, and M. I. Jordan. “Latent Dirichlet Allocation”. In: *Journal of machine Learning research* 3.Jan (2003), pp. 993–1022 (cit. on pp. 8, 20).

[21] J. Silva et al. “Using LocalMaxs Algorithm for the Extraction of Contiguous and Noncontiguous Multiword Lexical Units”. In: *EPIA’99* (Sept. 1999), pp. 113–132. DOI: 10.1007/3-540-48159-1_9 (cit. on pp. 9, 19).

[22] *"K-Nearest Neighbors visualization"*. URL: https://www.datacamp.com/tutorial/k-nearest-neighbor-classification-scikit-learn. (accessed: 04.09.2022) (cit. on p. 11).

[23] T. Joachims. "Text categorization with support vector machines: Learning with many relevant features". In: *European conference on machine learning*. Springer. 1998, pp. 137–142 (cit. on pp. 11, 20).

[24] R. Quinlan. "Induction of Decision Trees". In: *Machine Learning* 1.1 (Mar. 1986), pp. 81–106. DOI: 10.1007/BF00116251 (cit. on p. 12).

[25] T. Joachims. *A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization*. Tech. rep. Carnegie-mellon Univ Pittsburgh pa dept of computer science, 1996 (cit. on p. 12).

[26] O. I. Abiodun et al. "State-of-the-art in artificial neural network applications: A survey". In: *Heliyon* 4.11 (2018), e00938. ISSN: 2405-8440. DOI: https://doi.org/10.1016/j.heliyon.2018.e00938. URL: https://www.sciencedirect.com/science/article/pii/S2405844018332067 (cit. on p. 13).

[27] P. Rousseeuw. "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis". In: *Journal of Computational and Applied Mathematics* 20 (1987), pp. 53–65. DOI: 10.1016/0377-0427(87)90125-7 (cit. on p. 14).

[28] T. Zhang, R. Ramakrishnan, and M. Livny. "BIRCH: An Efficient Data Clustering Method for Very Large Databases". In: *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*. Vol. 25. SIGMOD '96 2. Montreal, Quebec, Canada: Association for Computing Machinery, 1996, pp. 103–114. ISBN: 0897917944. DOI: 10.1145/233269.233324. URL: https://doi.org/10.1145/233269.233324 (cit. on p. 15).

[29] M. Ester et al. "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. Vol. 96. AAAI Press, Jan. 1996, pp. 226–231 (cit. on p. 16).

[30] R. Campello et al. "Hierarchical Density Estimates for Data Clustering, Visualization, and Outlier Detection". In: *ACM Transactions on Knowledge Discovery from Data* 10 (July 2015), pp. 1–51. DOI: 10.1145/2733381 (cit. on p. 16).

[31] C. Aggarwal and C. Zhai. "A Survey of Text Clustering Algorithms". In: *Mining Text Data* (Aug. 2012), pp. 77–128. DOI: 10.1007/978-1-4614-3223-4_4 (cit. on p. 17).

[32] B. Frey and D. Dueck. "Clustering by Passing Messages Between Data Points". In: *Science (New York, N.Y.)* 315 (Mar. 2007), pp. 972–976. DOI: 10.1126/science.1136800 (cit. on p. 18).

[33] G. Forman. "An extensive empirical study of feature selection metrics for text classification". In: *Journal of Machine Learning Research - JMLR* 3 (Mar. 2003), pp. 1289–1305 (cit. on p. 19).

[34] J. Silva and G. Lopes. "A Local Maxima method and a Fair Dispersion Normalization for extracting multi-word units from corpora". In: (Jan. 1999), pp. 369–381 (cit. on p. 19).

[35] Y. Li and A. Jain. "Classification of text documents". In: *Proceedings. Fourteenth International Conference on Pattern Recognition*. Vol. 2. 1998, pp. 1295–1297. ISBN: 0-8186-8512-3. DOI: 10.1109/ICPR.1998.711938 (cit. on p. 20).

[36] B. Nigam et al. "Document Classification Using Expectation Maximization with Semi Supervised Learning". In: *International Journal on Soft Computing* 2.4 (Nov. 2011), pp. 37–44. DOI: 10.5121/ijsc.2011.2404 (cit. on p. 21).

[37] Y. Ko and J. See. "Automatic Text Categorization by Unsupervised Learning". In: (July 2000), pp. 453–459. DOI: 10.3115/990820.990886 (cit. on p. 22).

[38] *"The 5 Clustering Algorithms Data Scientists Need to Know"*. URL: https://towardsdatascience.com/the-5-clustering-algorithms-data-scientists-need-to-know-a36d136ef68. (accessed: 05.02.2022) (cit. on p. 22).

[39] *"Benchmarking Performance and Scaling of Python Clustering Algorithms"*. URL: https://hdbscan.readthedocs.io/en/latest/performance_and_scalability.html. (accessed: 05.02.2022) (cit. on p. 23).

[40] *"Clustering Performance Overview"*. URL: https://scikit-learn.org/stable/modules/clustering.html#clustering-performance-evaluation. (accessed: 05.02.2022) (cit. on p. 23).

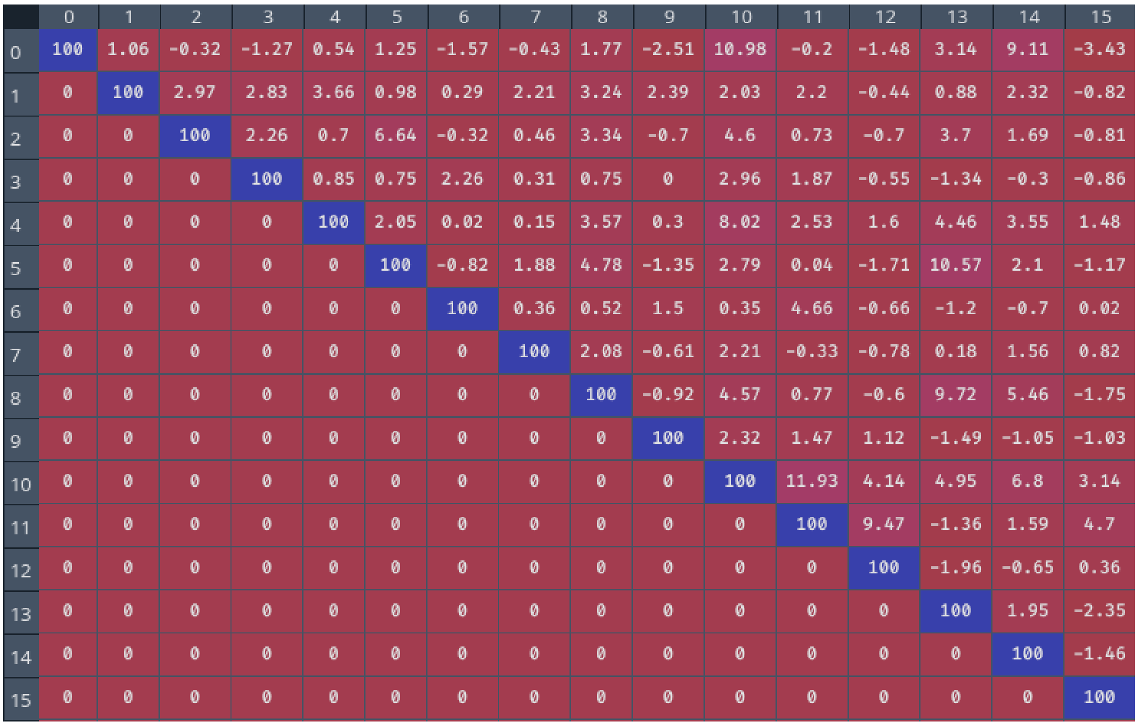
[41] R. García et al. "Comparison of Clustering Algorithms in Text Clustering Tasks". In: *Computación y Sistemas* 24 (June 2020), pp. 429–437. DOI: 10.13053/cys-24-2-3369 (cit. on p. 23).

[42] L. van der Maaten and G. E. Hinton. "Visualizing Data using t-SNE". In: *Journal of Machine Learning Research* 9 (Nov. 2008), pp. 2579–2605 (cit. on p. 24).

[43] *"Natural Language Toolkit"*. URL: https://www.nltk.org/index.html. (accessed: 04.09.2022) (cit. on p. 26).

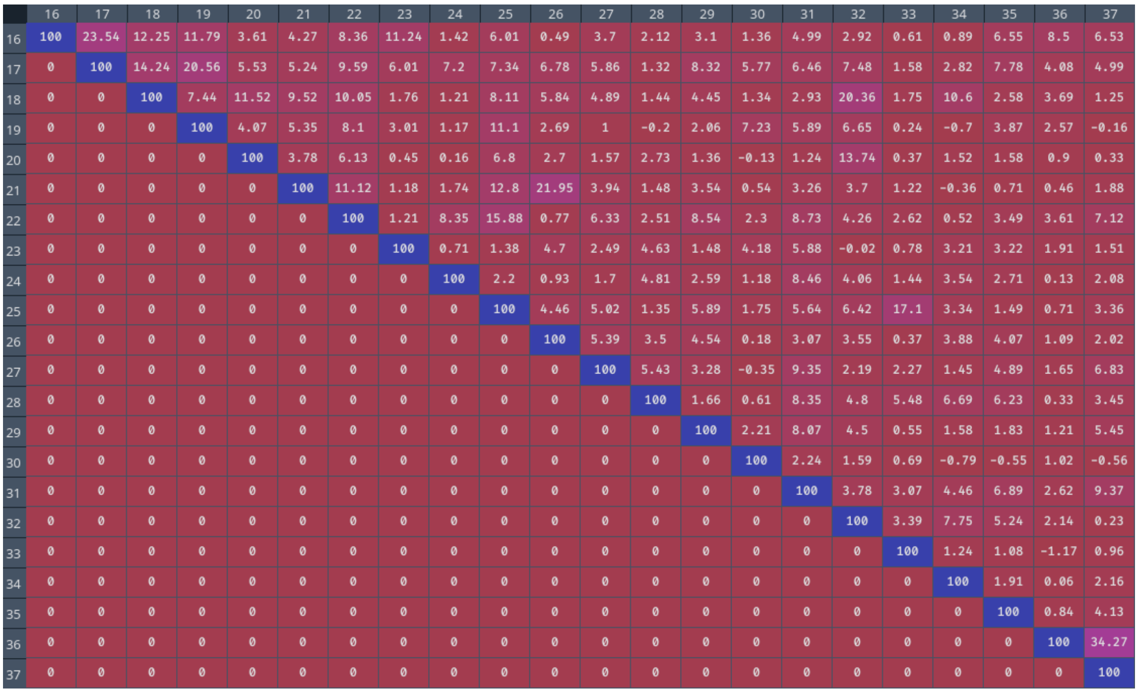
A

Variation Coefficient matrices



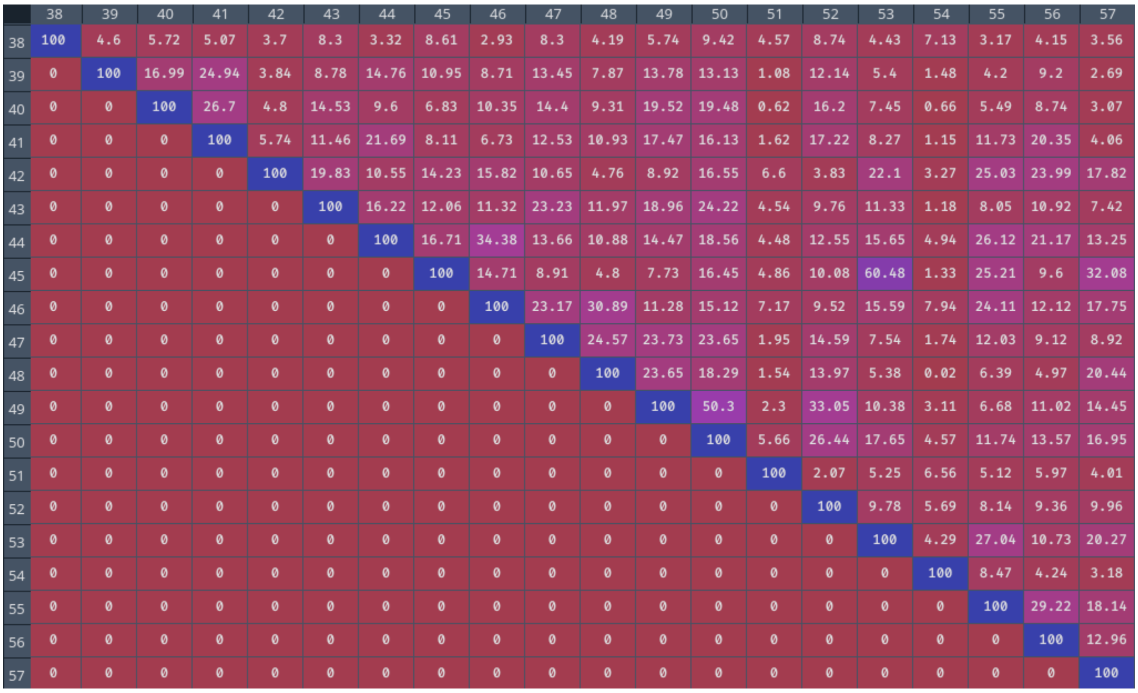
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	100	1.06	-0.32	-1.27	0.54	1.25	-1.57	-0.43	1.77	-2.51	10.98	-0.2	-1.48	3.14	9.11	-3.43
1	0	100	2.97	2.83	3.66	0.98	0.29	2.21	3.24	2.39	2.03	2.2	-0.44	0.88	2.32	-0.82
2	0	0	100	2.26	0.7	6.64	-0.32	0.46	3.34	-0.7	4.6	0.73	-0.7	3.7	1.69	-0.81
3	0	0	0	100	0.85	0.75	2.26	0.31	0.75	0	2.96	1.87	-0.55	-1.34	-0.3	-0.86
4	0	0	0	0	100	2.05	0.02	0.15	3.57	0.3	8.02	2.53	1.6	4.46	3.55	1.48
5	0	0	0	0	0	100	-0.82	1.88	4.78	-1.35	2.79	0.04	-1.71	10.57	2.1	-1.17
6	0	0	0	0	0	0	100	0.36	0.52	1.5	0.35	4.66	-0.66	-1.2	-0.7	0.02
7	0	0	0	0	0	0	0	100	2.08	-0.61	2.21	-0.33	-0.78	0.18	1.56	0.82
8	0	0	0	0	0	0	0	0	100	-0.92	4.57	0.77	-0.6	9.72	5.46	-1.75
9	0	0	0	0	0	0	0	0	0	100	2.32	1.47	1.12	-1.49	-1.05	-1.03
10	0	0	0	0	0	0	0	0	0	0	100	11.93	4.14	4.95	6.8	3.14
11	0	0	0	0	0	0	0	0	0	0	0	100	9.47	-1.36	1.59	4.7
12	0	0	0	0	0	0	0	0	0	0	0	0	100	-1.96	-0.65	0.36
13	0	0	0	0	0	0	0	0	0	0	0	0	0	100	1.95	-2.35
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	-1.46
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Figure A.1: Variation Coefficient matrix subsection of “Chemicals” documents (*corpus* I).



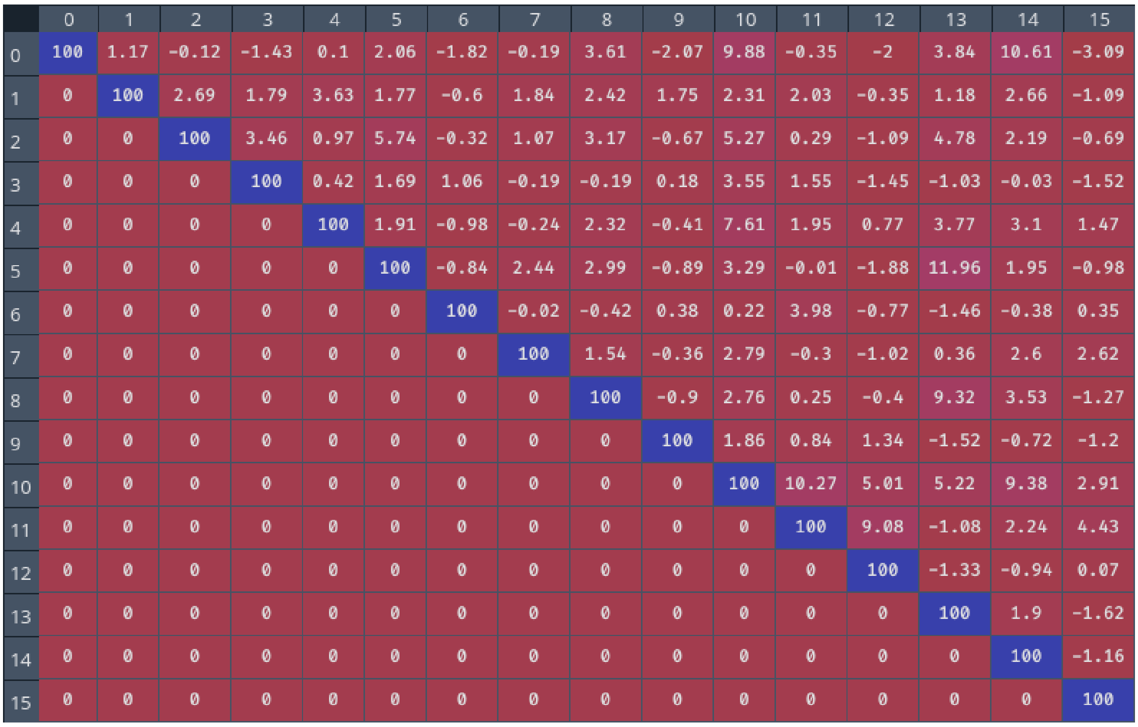
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
16	100	23.54	12.25	11.79	3.61	4.27	8.36	11.24	1.42	6.01	0.49	3.7	2.12	3.1	1.36	4.99	2.92	0.61	0.89	6.55	8.5	6.53
17	0	100	14.24	20.56	5.53	5.24	9.59	6.01	7.2	7.34	6.78	5.86	1.32	8.32	5.77	6.46	7.48	1.58	2.82	7.78	4.08	4.99
18	0	0	100	7.44	11.52	9.52	10.05	1.76	1.21	8.11	5.84	4.89	1.44	4.45	1.34	2.93	20.36	1.75	10.6	2.58	3.69	1.25
19	0	0	0	100	4.07	5.35	8.1	3.01	1.17	11.1	2.69	1	-0.2	2.06	7.23	5.89	6.65	0.24	-0.7	3.87	2.57	-0.16
20	0	0	0	0	100	3.78	6.13	0.45	0.16	6.8	2.7	1.57	2.73	1.36	-0.13	1.24	13.74	0.37	1.52	1.58	0.9	0.33
21	0	0	0	0	0	100	11.12	1.18	1.74	12.8	21.95	3.94	1.48	3.54	0.54	3.26	3.7	1.22	-0.36	0.71	0.46	1.88
22	0	0	0	0	0	0	100	1.21	8.35	15.88	0.77	6.33	2.51	8.54	2.3	8.73	4.26	2.62	0.52	3.49	3.61	7.12
23	0	0	0	0	0	0	0	100	0.71	1.38	4.7	2.49	4.63	1.48	4.18	5.88	-0.02	0.78	3.21	3.22	1.91	1.51
24	0	0	0	0	0	0	0	0	100	2.2	0.93	1.7	4.81	2.59	1.18	8.46	4.06	1.44	3.54	2.71	0.13	2.08
25	0	0	0	0	0	0	0	0	0	100	4.46	5.02	1.35	5.89	1.75	5.64	6.42	17.1	3.34	1.49	0.71	3.36
26	0	0	0	0	0	0	0	0	0	0	100	5.39	3.5	4.54	0.18	3.07	3.55	0.37	3.88	4.07	1.09	2.02
27	0	0	0	0	0	0	0	0	0	0	0	100	5.43	3.28	-0.35	9.35	2.19	2.27	1.45	4.89	1.65	6.83
28	0	0	0	0	0	0	0	0	0	0	0	0	100	1.66	0.61	8.35	4.8	5.48	6.69	6.23	0.33	3.45
29	0	0	0	0	0	0	0	0	0	0	0	0	0	100	2.21	8.07	4.5	0.55	1.58	1.83	1.21	5.45
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	2.24	1.59	0.69	-0.79	-0.55	1.02	-0.56
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	3.78	3.07	4.46	6.89	2.62	9.37
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	3.39	7.75	5.24	2.14	0.23
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	1.24	1.08	-1.17	0.96
34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	1.91	0.06	2.16
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0.84	4.13
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	34.27
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Figure A.2: Variation Coefficient matrix subsection of “Constellations” documents (*corpus* I).



	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57
38	100	4.6	5.72	5.07	3.7	8.3	3.32	8.61	2.93	8.3	4.19	5.74	9.42	4.57	8.74	4.43	7.13	3.17	4.15	3.56
39	0	100	16.99	24.94	3.84	8.78	14.76	10.95	8.71	13.45	7.87	13.78	13.13	1.08	12.14	5.4	1.48	4.2	9.2	2.69
40	0	0	100	26.7	4.8	14.53	9.6	6.83	10.35	14.4	9.31	19.52	19.48	0.62	16.2	7.45	0.66	5.49	8.74	3.07
41	0	0	0	100	5.74	11.46	21.69	8.11	6.73	12.53	10.93	17.47	16.13	1.62	17.22	8.27	1.15	11.73	20.35	4.06
42	0	0	0	0	100	19.83	10.55	14.23	15.82	10.65	4.76	8.92	16.55	6.6	3.83	22.1	3.27	25.03	23.99	17.82
43	0	0	0	0	0	100	16.22	12.06	11.32	23.23	11.97	18.96	24.22	4.54	9.76	11.33	1.18	8.05	10.92	7.42
44	0	0	0	0	0	0	100	16.71	34.38	13.66	10.88	14.47	18.56	4.48	12.55	15.65	4.94	26.12	21.17	13.25
45	0	0	0	0	0	0	0	100	14.71	8.91	4.8	7.73	16.45	4.86	10.08	60.48	1.33	25.21	9.6	32.08
46	0	0	0	0	0	0	0	0	100	23.17	30.89	11.28	15.12	7.17	9.52	15.59	7.94	24.11	12.12	17.75
47	0	0	0	0	0	0	0	0	0	100	24.57	23.73	23.65	1.95	14.59	7.54	1.74	12.03	9.12	8.92
48	0	0	0	0	0	0	0	0	0	0	100	23.65	18.29	1.54	13.97	5.38	0.02	6.39	4.97	20.44
49	0	0	0	0	0	0	0	0	0	0	0	100	50.3	2.3	33.05	10.38	3.11	6.68	11.02	14.45
50	0	0	0	0	0	0	0	0	0	0	0	0	100	5.66	26.44	17.65	4.57	11.74	13.57	16.95
51	0	0	0	0	0	0	0	0	0	0	0	0	0	100	2.07	5.25	6.56	5.12	5.97	4.01
52	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	9.78	5.69	8.14	9.36	9.96
53	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	4.29	27.04	10.73	20.27
54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	8.47	4.24	3.18
55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	29.22	18.14
56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	12.96
57	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Figure A.3: Variation Coefficient matrix subsection of “Tennis” documents (*corpus* I).

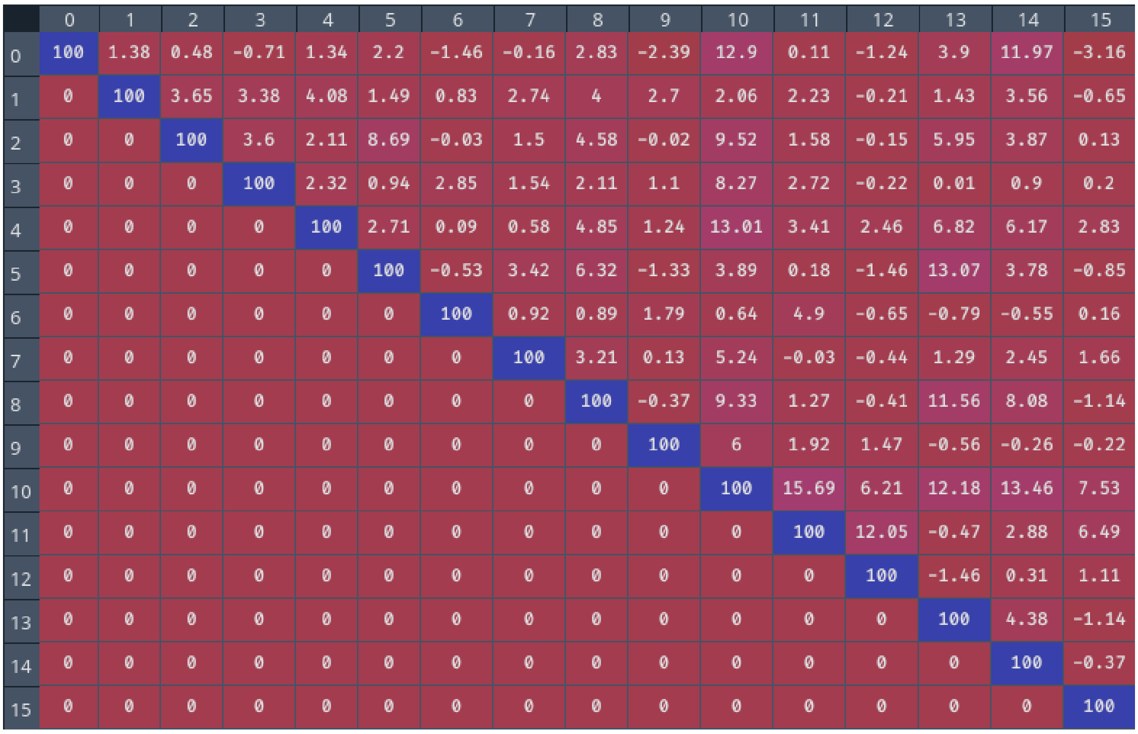


	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	100	1.17	-0.12	-1.43	0.1	2.06	-1.82	-0.19	3.61	-2.07	9.88	-0.35	-2	3.84	10.61	-3.09
1	0	100	2.69	1.79	3.63	1.77	-0.6	1.84	2.42	1.75	2.31	2.03	-0.35	1.18	2.66	-1.09
2	0	0	100	3.46	0.97	5.74	-0.32	1.07	3.17	-0.67	5.27	0.29	-1.09	4.78	2.19	-0.69
3	0	0	0	100	0.42	1.69	1.06	-0.19	-0.19	0.18	3.55	1.55	-1.45	-1.03	-0.03	-1.52
4	0	0	0	0	100	1.91	-0.98	-0.24	2.32	-0.41	7.61	1.95	0.77	3.77	3.1	1.47
5	0	0	0	0	0	100	-0.84	2.44	2.99	-0.89	3.29	-0.01	-1.88	11.96	1.95	-0.98
6	0	0	0	0	0	0	100	-0.02	-0.42	0.38	0.22	3.98	-0.77	-1.46	-0.38	0.35
7	0	0	0	0	0	0	0	100	1.54	-0.36	2.79	-0.3	-1.02	0.36	2.6	2.62
8	0	0	0	0	0	0	0	0	100	-0.9	2.76	0.25	-0.4	9.32	3.53	-1.27
9	0	0	0	0	0	0	0	0	0	100	1.86	0.84	1.34	-1.52	-0.72	-1.2
10	0	0	0	0	0	0	0	0	0	0	100	10.27	5.01	5.22	9.38	2.91
11	0	0	0	0	0	0	0	0	0	0	0	100	9.08	-1.08	2.24	4.43
12	0	0	0	0	0	0	0	0	0	0	0	0	100	-1.33	-0.94	0.07
13	0	0	0	0	0	0	0	0	0	0	0	0	0	100	1.9	-1.62
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	-1.16
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Figure A.4: Variation Coefficient matrix subsection of “Constellations” documents, with the inclusion of the Median metric (*corpus* I).

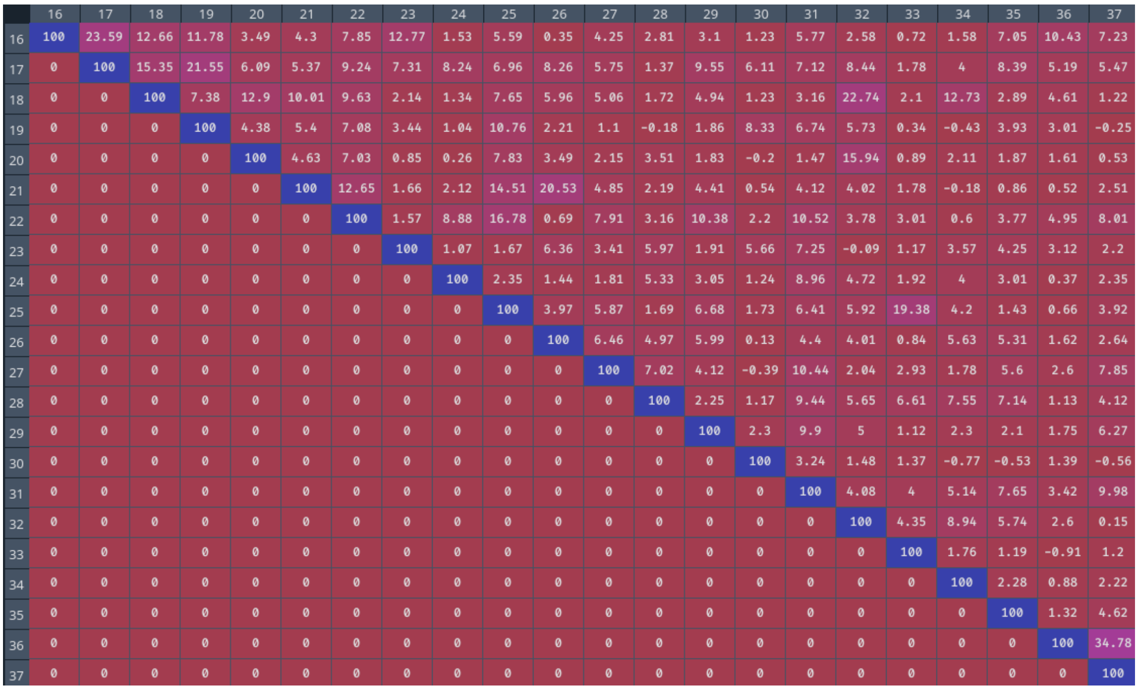
B

Skewness Coefficient (3rd moment) matrices



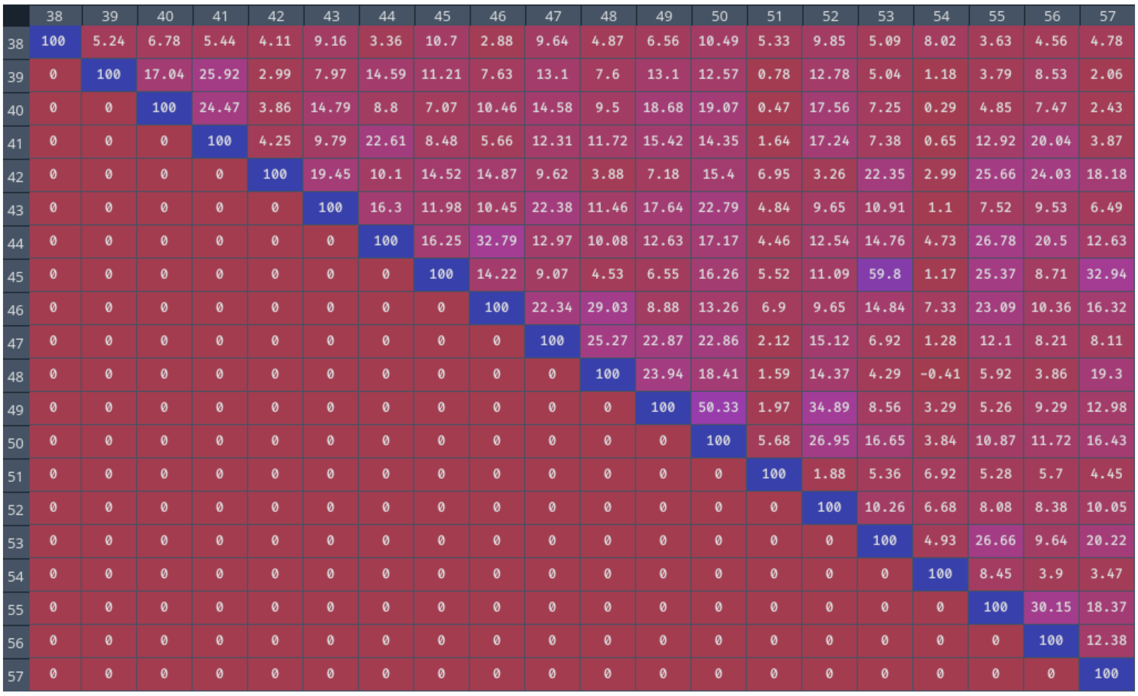
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	100	1.38	0.48	-0.71	1.34	2.2	-1.46	-0.16	2.83	-2.39	12.9	0.11	-1.24	3.9	11.97	-3.16
1	0	100	3.65	3.38	4.08	1.49	0.83	2.74	4	2.7	2.06	2.23	-0.21	1.43	3.56	-0.65
2	0	0	100	3.6	2.11	8.69	-0.03	1.5	4.58	-0.02	9.52	1.58	-0.15	5.95	3.87	0.13
3	0	0	0	100	2.32	0.94	2.85	1.54	2.11	1.1	8.27	2.72	-0.22	0.01	0.9	0.2
4	0	0	0	0	100	2.71	0.09	0.58	4.85	1.24	13.01	3.41	2.46	6.82	6.17	2.83
5	0	0	0	0	0	100	-0.53	3.42	6.32	-1.33	3.89	0.18	-1.46	13.07	3.78	-0.85
6	0	0	0	0	0	0	100	0.92	0.89	1.79	0.64	4.9	-0.65	-0.79	-0.55	0.16
7	0	0	0	0	0	0	0	100	3.21	0.13	5.24	-0.03	-0.44	1.29	2.45	1.66
8	0	0	0	0	0	0	0	0	100	-0.37	9.33	1.27	-0.41	11.56	8.08	-1.14
9	0	0	0	0	0	0	0	0	0	100	6	1.92	1.47	-0.56	-0.26	-0.22
10	0	0	0	0	0	0	0	0	0	0	100	15.69	6.21	12.18	13.46	7.53
11	0	0	0	0	0	0	0	0	0	0	0	100	12.05	-0.47	2.88	6.49
12	0	0	0	0	0	0	0	0	0	0	0	0	100	-1.46	0.31	1.11
13	0	0	0	0	0	0	0	0	0	0	0	0	0	100	4.38	-1.14
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	-0.37
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Figure B.1: Skewness matrix subsection of "Chemicals" documents (*corpus* I).



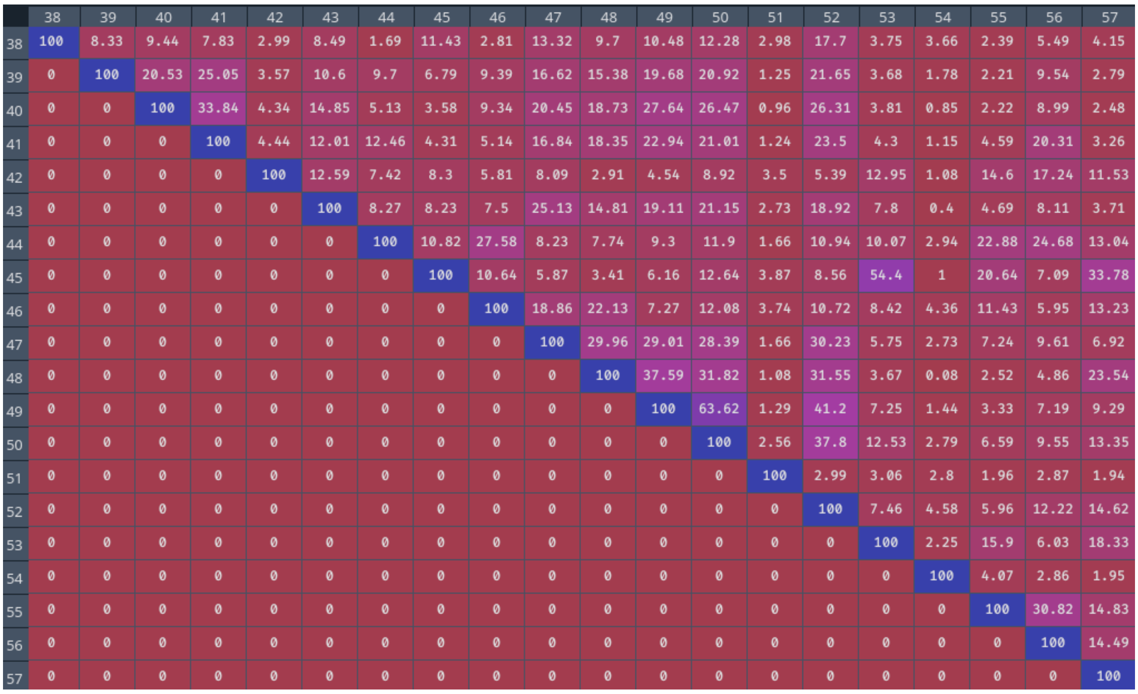
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
16	100	23.59	12.66	11.78	3.49	4.3	7.85	12.77	1.53	5.59	0.35	4.25	2.81	3.1	1.23	5.77	2.58	0.72	1.58	7.05	10.43	7.23
17	0	100	15.35	21.55	6.09	5.37	9.24	7.31	8.24	6.96	8.26	5.75	1.37	9.55	6.11	7.12	8.44	1.78	4	8.39	5.19	5.47
18	0	0	100	7.38	12.9	10.01	9.63	2.14	1.34	7.65	5.96	5.06	1.72	4.94	1.23	3.16	22.74	2.1	12.73	2.89	4.61	1.22
19	0	0	0	100	4.38	5.4	7.08	3.44	1.04	10.76	2.21	1.1	-0.18	1.86	8.33	6.74	5.73	0.34	-0.43	3.93	3.01	-0.25
20	0	0	0	0	100	4.63	7.03	0.85	0.26	7.83	3.49	2.15	3.51	1.83	-0.2	1.47	15.94	0.89	2.11	1.87	1.61	0.53
21	0	0	0	0	0	100	12.65	1.66	2.12	14.51	20.53	4.85	2.19	4.41	0.54	4.12	4.02	1.78	-0.18	0.86	0.52	2.51
22	0	0	0	0	0	0	100	1.57	8.88	16.78	0.69	7.91	3.16	10.38	2.2	10.52	3.78	3.01	0.6	3.77	4.95	8.01
23	0	0	0	0	0	0	0	100	1.07	1.67	6.36	3.41	5.97	1.91	5.66	7.25	-0.09	1.17	3.57	4.25	3.12	2.2
24	0	0	0	0	0	0	0	0	100	2.35	1.44	1.81	5.33	3.05	1.24	8.96	4.72	1.92	4	3.01	0.37	2.35
25	0	0	0	0	0	0	0	0	0	100	3.97	5.87	1.69	6.68	1.73	6.41	5.92	19.38	4.2	1.43	0.66	3.92
26	0	0	0	0	0	0	0	0	0	0	100	6.46	4.97	5.99	0.13	4.4	4.01	0.84	5.63	5.31	1.62	2.64
27	0	0	0	0	0	0	0	0	0	0	0	100	7.02	4.12	-0.39	10.44	2.04	2.93	1.78	5.6	2.6	7.85
28	0	0	0	0	0	0	0	0	0	0	0	0	100	2.25	1.17	9.44	5.65	6.61	7.55	7.14	1.13	4.12
29	0	0	0	0	0	0	0	0	0	0	0	0	0	100	2.3	9.9	5	1.12	2.3	2.1	1.75	6.27
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	3.24	1.48	1.37	-0.77	-0.53	1.39	-0.56
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	4.08	4	5.14	7.65	3.42	9.98
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	4.35	8.94	5.74	2.6	0.15
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	1.76	1.19	-0.91	1.2
34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	2.28	0.88	2.22
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	1.32	4.62
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	34.78
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Figure B.2: Skewness matrix subsection of “Constellations” documents (*corpus* I).



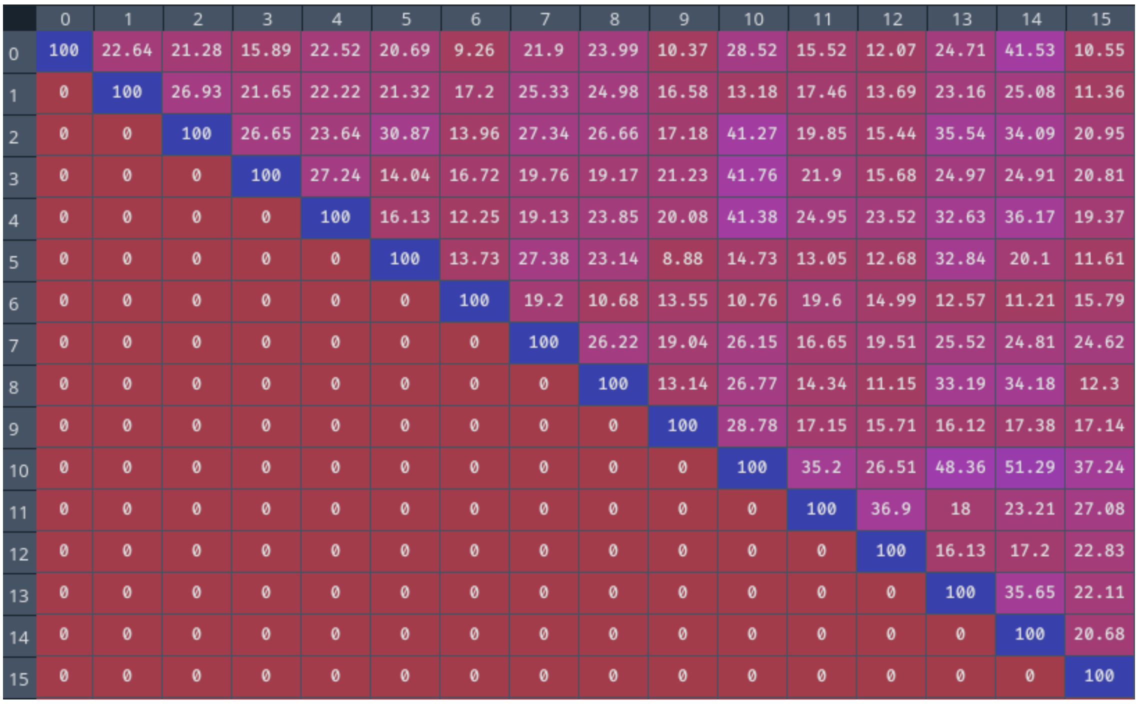
	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57
38	100	5.24	6.78	5.44	4.11	9.16	3.36	10.7	2.88	9.64	4.87	6.56	10.49	5.33	9.85	5.09	8.02	3.63	4.56	4.78
39	0	100	17.04	25.92	2.99	7.97	14.59	11.21	7.63	13.1	7.6	13.1	12.57	0.78	12.78	5.04	1.18	3.79	8.53	2.06
40	0	0	100	24.47	3.86	14.79	8.8	7.07	10.46	14.58	9.5	18.68	19.07	0.47	17.56	7.25	0.29	4.85	7.47	2.43
41	0	0	0	100	4.25	9.79	22.61	8.48	5.66	12.31	11.72	15.42	14.35	1.64	17.24	7.38	0.65	12.92	20.04	3.87
42	0	0	0	0	100	19.45	10.1	14.52	14.87	9.62	3.88	7.18	15.4	6.95	3.26	22.35	2.99	25.66	24.03	18.18
43	0	0	0	0	0	100	16.3	11.98	10.45	22.38	11.46	17.64	22.79	4.84	9.65	10.91	1.1	7.52	9.53	6.49
44	0	0	0	0	0	0	100	16.25	32.79	12.97	10.08	12.63	17.17	4.46	12.54	14.76	4.73	26.78	20.5	12.63
45	0	0	0	0	0	0	0	100	14.22	9.07	4.53	6.55	16.26	5.52	11.09	59.8	1.17	25.37	8.71	32.94
46	0	0	0	0	0	0	0	0	100	22.34	29.03	8.88	13.26	6.9	9.65	14.84	7.33	23.09	10.36	16.32
47	0	0	0	0	0	0	0	0	0	100	25.27	22.87	22.86	2.12	15.12	6.92	1.28	12.1	8.21	8.11
48	0	0	0	0	0	0	0	0	0	0	100	23.94	18.41	1.59	14.37	4.29	-0.41	5.92	3.86	19.3
49	0	0	0	0	0	0	0	0	0	0	0	100	50.33	1.97	34.89	8.56	3.29	5.26	9.29	12.98
50	0	0	0	0	0	0	0	0	0	0	0	0	100	5.68	26.95	16.65	3.84	10.87	11.72	16.43
51	0	0	0	0	0	0	0	0	0	0	0	0	0	100	1.88	5.36	6.92	5.28	5.7	4.45
52	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	10.26	6.68	8.08	8.38	10.05
53	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	4.93	26.66	9.64	20.22
54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	8.45	3.9	3.47
55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	30.15	18.37
56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	12.38
57	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Figure B.3: Skewness matrix subsection of “Tennis” documents (*corpus* I).



	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57
38	100	8.33	9.44	7.83	2.99	8.49	1.69	11.43	2.81	13.32	9.7	10.48	12.28	2.98	17.7	3.75	3.66	2.39	5.49	4.15
39	0	100	20.53	25.05	3.57	10.6	9.7	6.79	9.39	16.62	15.38	19.68	20.92	1.25	21.65	3.68	1.78	2.21	9.54	2.79
40	0	0	100	33.84	4.34	14.85	5.13	3.58	9.34	20.45	18.73	27.64	26.47	0.96	26.31	3.81	0.85	2.22	8.99	2.48
41	0	0	0	100	4.44	12.01	12.46	4.31	5.14	16.84	18.35	22.94	21.01	1.24	23.5	4.3	1.15	4.59	20.31	3.26
42	0	0	0	0	100	12.59	7.42	8.3	5.81	8.09	2.91	4.54	8.92	3.5	5.39	12.95	1.08	14.6	17.24	11.53
43	0	0	0	0	0	100	8.27	8.23	7.5	25.13	14.81	19.11	21.15	2.73	18.92	7.8	0.4	4.69	8.11	3.71
44	0	0	0	0	0	0	100	10.82	27.58	8.23	7.74	9.3	11.9	1.66	10.94	10.07	2.94	22.88	24.68	13.04
45	0	0	0	0	0	0	0	100	10.64	5.87	3.41	6.16	12.64	3.87	8.56	54.4	1	20.64	7.09	33.78
46	0	0	0	0	0	0	0	0	100	18.86	22.13	7.27	12.08	3.74	10.72	8.42	4.36	11.43	5.95	13.23
47	0	0	0	0	0	0	0	0	0	100	29.96	29.01	28.39	1.66	30.23	5.75	2.73	7.24	9.61	6.92
48	0	0	0	0	0	0	0	0	0	0	100	37.59	31.82	1.08	31.55	3.67	0.08	2.52	4.86	23.54
49	0	0	0	0	0	0	0	0	0	0	0	100	63.62	1.29	41.2	7.25	1.44	3.33	7.19	9.29
50	0	0	0	0	0	0	0	0	0	0	0	0	100	2.56	37.8	12.53	2.79	6.59	9.55	13.35
51	0	0	0	0	0	0	0	0	0	0	0	0	0	100	2.99	3.06	2.8	1.96	2.87	1.94
52	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	7.46	4.58	5.96	12.22	14.62
53	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	2.25	15.9	6.03	18.33
54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	4.07	2.86	1.95
55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	30.82	14.83
56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	14.49
57	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Figure B.4: Skewness matrix subsection of "Tennis" documents, with the inclusion of the Median metric (*corpus* I).

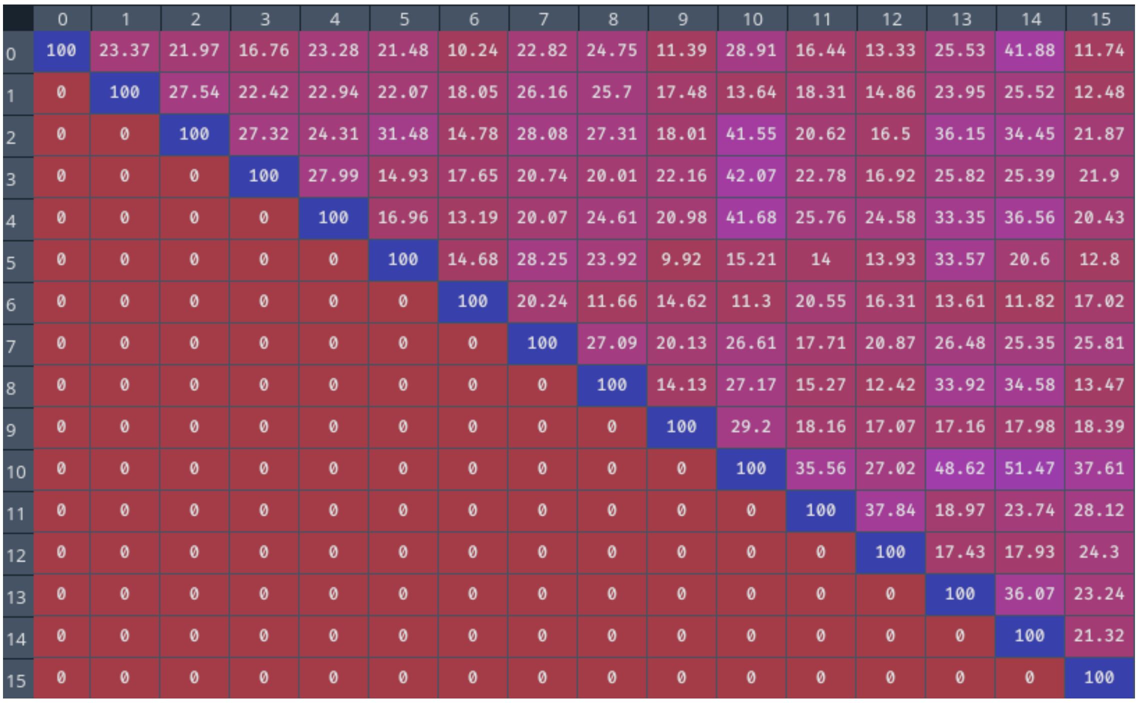


	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	100	22.64	21.28	15.89	22.52	20.69	9.26	21.9	23.99	10.37	28.52	15.52	12.07	24.71	41.53	10.55
1	0	100	26.93	21.65	22.22	21.32	17.2	25.33	24.98	16.58	13.18	17.46	13.69	23.16	25.08	11.36
2	0	0	100	26.65	23.64	30.87	13.96	27.34	26.66	17.18	41.27	19.85	15.44	35.54	34.09	20.95
3	0	0	0	100	27.24	14.04	16.72	19.76	19.17	21.23	41.76	21.9	15.68	24.97	24.91	20.81
4	0	0	0	0	100	16.13	12.25	19.13	23.85	20.08	41.38	24.95	23.52	32.63	36.17	19.37
5	0	0	0	0	0	100	13.73	27.38	23.14	8.88	14.73	13.05	12.68	32.84	20.1	11.61
6	0	0	0	0	0	0	100	19.2	10.68	13.55	10.76	19.6	14.99	12.57	11.21	15.79
7	0	0	0	0	0	0	0	100	26.22	19.04	26.15	16.65	19.51	25.52	24.81	24.62
8	0	0	0	0	0	0	0	0	100	13.14	26.77	14.34	11.15	33.19	34.18	12.3
9	0	0	0	0	0	0	0	0	0	100	28.78	17.15	15.71	16.12	17.38	17.14
10	0	0	0	0	0	0	0	0	0	0	100	35.2	26.51	48.36	51.29	37.24
11	0	0	0	0	0	0	0	0	0	0	0	100	36.9	18	23.21	27.08
12	0	0	0	0	0	0	0	0	0	0	0	0	100	16.13	17.2	22.83
13	0	0	0	0	0	0	0	0	0	0	0	0	0	100	35.65	22.11
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	20.68
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Figure B.5: Skewness matrix subsection of "Chemicals" documents, with the inclusion of the *W* function. Worth noting that this subsection was the worst subsection of the matrices, when obtained through Skewness only (*corpus* I).

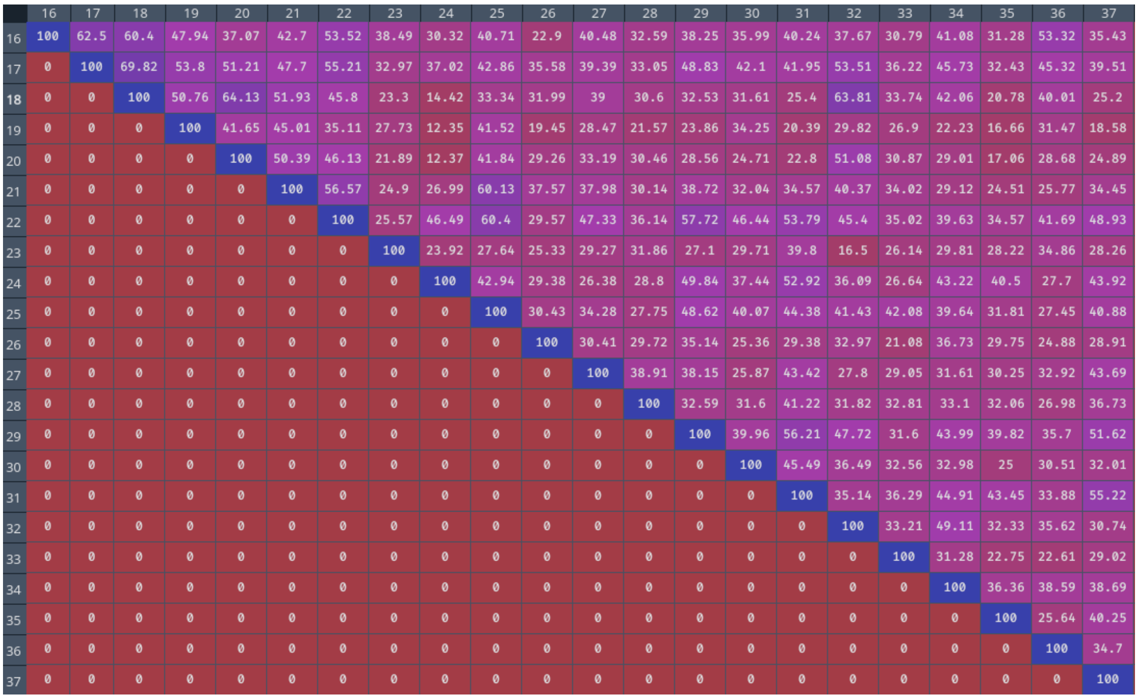
C

PSkJW matrices



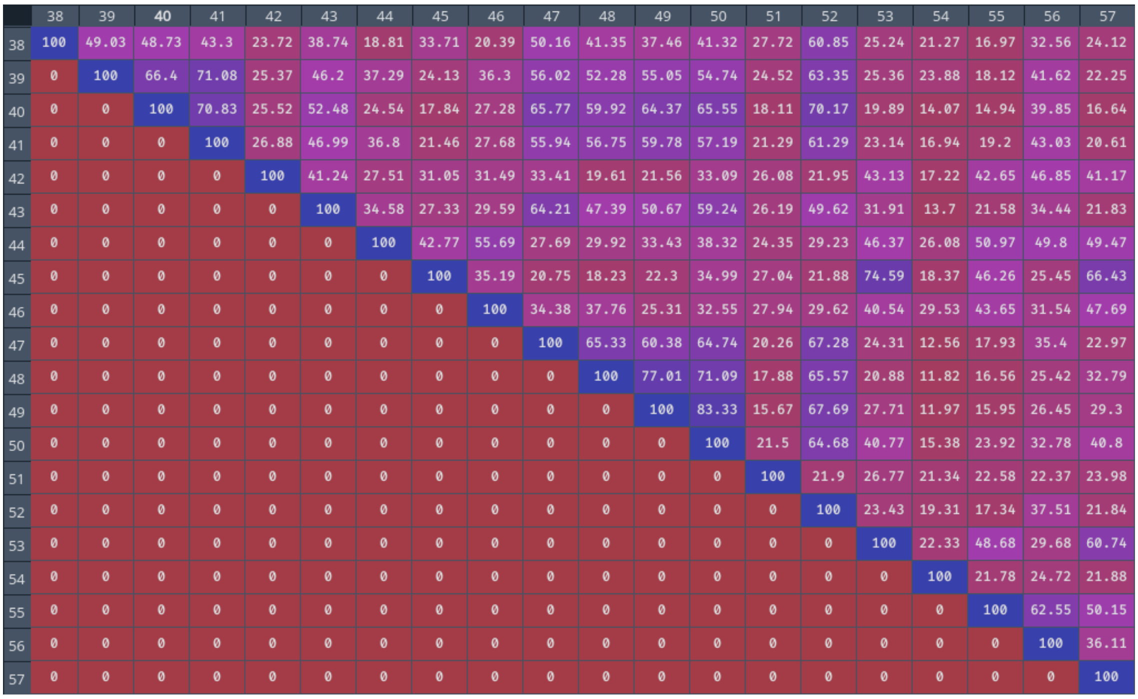
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	100	23.37	21.97	16.76	23.28	21.48	10.24	22.82	24.75	11.39	28.91	16.44	13.33	25.53	41.88	11.74
1	0	100	27.54	22.42	22.94	22.07	18.05	26.16	25.7	17.48	13.64	18.31	14.86	23.95	25.52	12.48
2	0	0	100	27.32	24.31	31.48	14.78	28.08	27.31	18.01	41.55	20.62	16.5	36.15	34.45	21.87
3	0	0	0	100	27.99	14.93	17.65	20.74	20.01	22.16	42.07	22.78	16.92	25.82	25.39	21.9
4	0	0	0	0	100	16.96	13.19	20.07	24.61	20.98	41.68	25.76	24.58	33.35	36.56	20.43
5	0	0	0	0	0	100	14.68	28.25	23.92	9.92	15.21	14	13.93	33.57	20.6	12.8
6	0	0	0	0	0	0	100	20.24	11.66	14.62	11.3	20.55	16.31	13.61	11.82	17.02
7	0	0	0	0	0	0	0	100	27.09	20.13	26.61	17.71	20.87	26.48	25.35	25.81
8	0	0	0	0	0	0	0	0	100	14.13	27.17	15.27	12.42	33.92	34.58	13.47
9	0	0	0	0	0	0	0	0	0	100	29.2	18.16	17.07	17.16	17.98	18.39
10	0	0	0	0	0	0	0	0	0	0	100	35.56	27.02	48.62	51.47	37.61
11	0	0	0	0	0	0	0	0	0	0	0	100	37.84	18.97	23.74	28.12
12	0	0	0	0	0	0	0	0	0	0	0	0	100	17.43	17.93	24.3
13	0	0	0	0	0	0	0	0	0	0	0	0	0	100	36.07	23.24
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	21.32
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Figure C.1: *PSkJW* matrix subsection of “Chemicals” documents (*corpus* I).



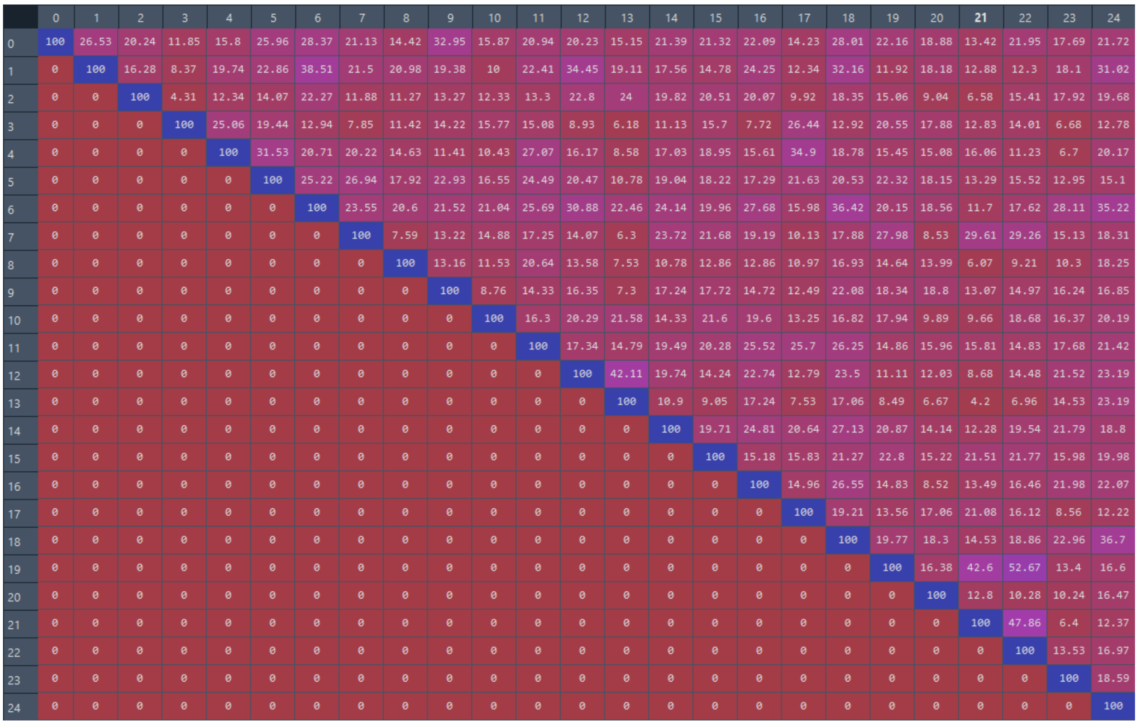
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
16	100	62.5	60.4	47.94	37.07	42.7	53.52	38.49	30.32	40.71	22.9	40.48	32.59	38.25	35.99	40.24	37.67	30.79	41.08	31.28	53.32	35.43
17	0	100	69.82	53.8	51.21	47.7	55.21	32.97	37.02	42.86	35.58	39.39	33.05	48.83	42.1	41.95	53.51	36.22	45.73	32.43	45.32	39.51
18	0	0	100	50.76	64.13	51.93	45.8	23.3	14.42	33.34	31.99	39	30.6	32.53	31.61	25.4	63.81	33.74	42.06	20.78	40.01	25.2
19	0	0	0	100	41.65	45.01	35.11	27.73	12.35	41.52	19.45	28.47	21.57	23.86	34.25	20.39	29.82	26.9	22.23	16.66	31.47	18.58
20	0	0	0	0	100	50.39	46.13	21.89	12.37	41.84	29.26	33.19	30.46	28.56	24.71	22.8	51.08	30.87	29.01	17.06	28.68	24.89
21	0	0	0	0	0	100	56.57	24.9	26.99	60.13	37.57	37.98	30.14	38.72	32.04	34.57	40.37	34.02	29.12	24.51	25.77	34.45
22	0	0	0	0	0	0	100	25.57	46.49	60.4	29.57	47.33	36.14	57.72	46.44	53.79	45.4	35.02	39.63	34.57	41.69	48.93
23	0	0	0	0	0	0	0	100	23.92	27.64	25.33	29.27	31.86	27.1	29.71	39.8	16.5	26.14	29.81	28.22	34.86	28.26
24	0	0	0	0	0	0	0	0	100	42.94	29.38	26.38	28.8	49.84	37.44	52.92	36.09	26.64	43.22	40.5	27.7	43.92
25	0	0	0	0	0	0	0	0	0	100	30.43	34.28	27.75	48.62	40.07	44.38	41.43	42.08	39.64	31.81	27.45	40.88
26	0	0	0	0	0	0	0	0	0	0	100	30.41	29.72	35.14	25.36	29.38	32.97	21.08	36.73	29.75	24.88	28.91
27	0	0	0	0	0	0	0	0	0	0	0	100	38.91	38.15	25.87	43.42	27.8	29.05	31.61	30.25	32.92	43.69
28	0	0	0	0	0	0	0	0	0	0	0	0	100	32.59	31.6	41.22	31.82	32.81	33.1	32.06	26.98	36.73
29	0	0	0	0	0	0	0	0	0	0	0	0	0	100	39.96	56.21	47.72	31.6	43.99	39.82	35.7	51.62
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	45.49	36.49	32.56	32.98	25	30.51	32.01
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	35.14	36.29	44.91	43.45	33.88	55.22
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	33.21	49.11	32.33	35.62	30.74
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	31.28	22.75	22.61	29.02
34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	36.36	38.59	38.69
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	25.64	40.25
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	34.7
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Figure C.2: *PSkJW* matrix subsection of “Constellations” documents (*corpus* I).



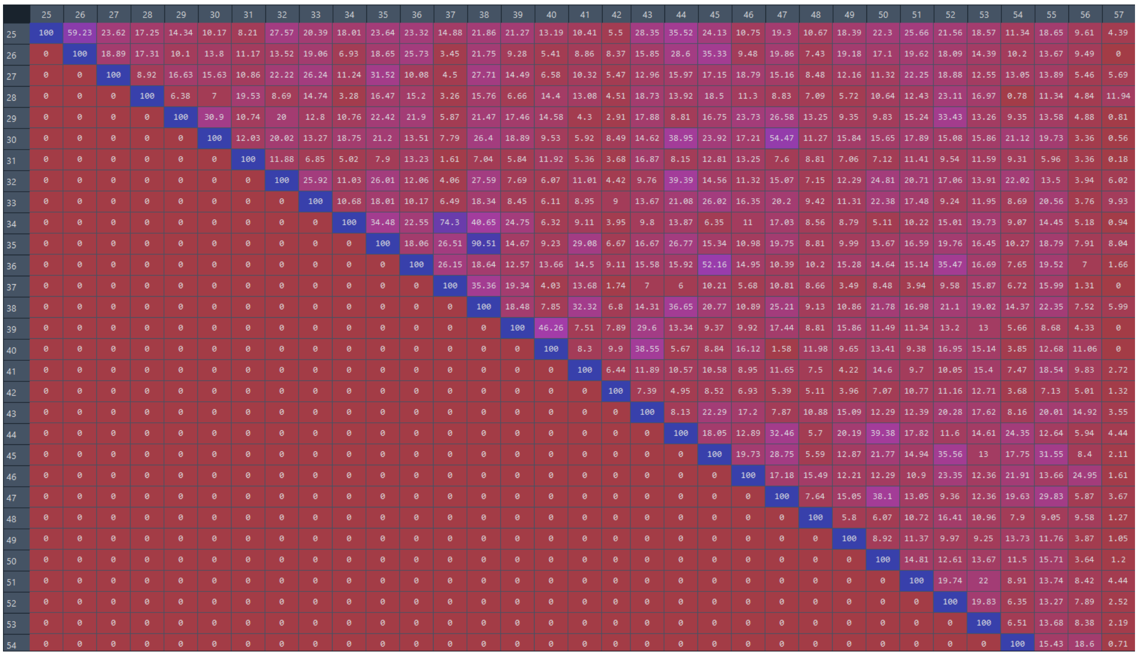
	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57
38	100	49.03	48.73	43.3	23.72	38.74	18.81	33.71	20.39	50.16	41.35	37.46	41.32	27.72	60.85	25.24	21.27	16.97	32.56	24.12
39	0	100	66.4	71.08	25.37	46.2	37.29	24.13	36.3	56.02	52.28	55.05	54.74	24.52	63.35	25.36	23.88	18.12	41.62	22.25
40	0	0	100	70.83	25.52	52.48	24.54	17.84	27.28	65.77	59.92	64.37	65.55	18.11	70.17	19.89	14.07	14.94	39.85	16.64
41	0	0	0	100	26.88	46.99	36.8	21.46	27.68	55.94	56.75	59.78	57.19	21.29	61.29	23.14	16.94	19.2	43.03	20.61
42	0	0	0	0	100	41.24	27.51	31.05	31.49	33.41	19.61	21.56	33.09	26.08	21.95	43.13	17.22	42.65	46.85	41.17
43	0	0	0	0	0	100	34.58	27.33	29.59	64.21	47.39	50.67	59.24	26.19	49.62	31.91	13.7	21.58	34.44	21.83
44	0	0	0	0	0	0	100	42.77	55.69	27.69	29.92	33.43	38.32	24.35	29.23	46.37	26.08	50.97	49.8	49.47
45	0	0	0	0	0	0	0	100	35.19	20.75	18.23	22.3	34.99	27.04	21.88	74.59	18.37	46.26	25.45	66.43
46	0	0	0	0	0	0	0	0	100	34.38	37.76	25.31	32.55	27.94	29.62	40.54	29.53	43.65	31.54	47.69
47	0	0	0	0	0	0	0	0	0	100	65.33	60.38	64.74	20.26	67.28	24.31	12.56	17.93	35.4	22.97
48	0	0	0	0	0	0	0	0	0	0	100	77.01	71.09	17.88	65.57	20.88	11.82	16.56	25.42	32.79
49	0	0	0	0	0	0	0	0	0	0	0	100	83.33	15.67	67.69	27.71	11.97	15.95	26.45	29.3
50	0	0	0	0	0	0	0	0	0	0	0	0	100	21.5	64.68	40.77	15.38	23.92	32.78	40.8
51	0	0	0	0	0	0	0	0	0	0	0	0	0	100	21.9	26.77	21.34	22.58	22.37	23.98
52	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	23.43	19.31	17.34	37.51	21.84
53	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	22.33	48.68	29.68	60.74
54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	21.78	24.72	21.88
55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	62.55	50.15
56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	36.11
57	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Figure C.3: *PSkJW* matrix subsection of “Tennis” documents (*corpus* I).



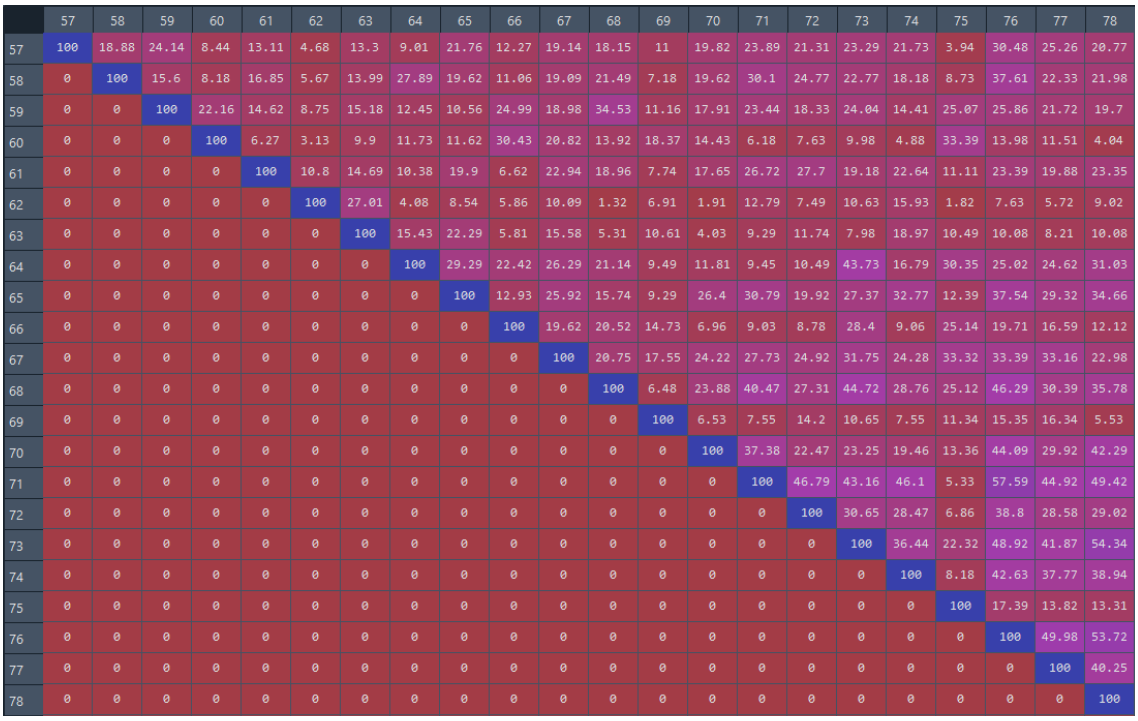
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
0	100	26.53	20.24	11.85	15.8	25.96	28.37	21.13	14.42	32.95	15.87	20.94	20.23	15.15	21.39	21.32	22.09	14.23	28.01	22.16	18.88	13.42	21.95	17.69	21.72
1	0	100	16.28	8.37	19.74	22.86	38.51	21.5	20.98	19.38	10	22.41	34.45	19.11	17.56	14.78	24.25	12.34	32.16	11.92	18.18	12.88	12.3	18.1	31.02
2	0	0	100	4.31	12.34	14.07	22.27	11.88	11.27	13.27	12.33	13.3	22.8	24	19.82	20.51	20.07	9.92	18.35	15.06	9.04	6.58	15.41	17.92	19.68
3	0	0	0	100	25.06	19.44	12.94	7.85	11.42	14.22	15.77	15.08	8.93	6.18	11.13	15.7	7.72	26.44	12.92	20.55	17.88	12.83	14.01	6.68	12.78
4	0	0	0	0	100	31.53	20.71	20.22	14.63	11.41	10.43	27.07	16.17	8.58	17.03	18.95	15.61	34.9	18.78	15.45	15.08	16.06	11.23	6.7	20.17
5	0	0	0	0	0	100	25.22	26.94	17.92	22.93	16.55	24.49	20.47	10.78	19.04	18.22	17.29	21.63	20.53	22.32	18.15	13.29	15.52	12.95	15.1
6	0	0	0	0	0	0	100	23.55	20.6	21.52	21.04	25.69	30.88	22.46	24.14	19.96	27.68	15.98	36.42	20.15	18.56	11.7	17.62	28.11	35.22
7	0	0	0	0	0	0	0	100	7.59	13.22	14.88	17.25	14.07	6.3	23.72	21.68	19.19	10.13	17.88	27.98	8.53	29.61	29.26	15.13	18.31
8	0	0	0	0	0	0	0	0	100	13.16	11.53	20.64	13.58	7.53	10.78	12.86	12.86	10.97	16.93	14.64	13.99	6.07	9.21	10.3	18.25
9	0	0	0	0	0	0	0	0	0	100	8.76	14.33	16.35	7.3	17.24	17.72	14.72	12.49	22.08	18.34	18.8	13.07	14.97	16.24	16.85
10	0	0	0	0	0	0	0	0	0	0	100	16.3	20.29	21.58	14.33	21.6	19.6	13.25	16.82	17.94	9.89	9.66	18.68	16.37	20.19
11	0	0	0	0	0	0	0	0	0	0	0	100	17.34	14.79	19.49	20.28	25.52	25.7	26.25	14.86	15.96	15.81	14.83	17.68	21.42
12	0	0	0	0	0	0	0	0	0	0	0	0	100	42.11	19.74	14.24	22.74	12.79	23.5	11.11	12.03	8.68	14.48	21.52	23.19
13	0	0	0	0	0	0	0	0	0	0	0	0	0	100	10.9	9.05	17.24	7.53	17.06	8.49	6.67	4.2	6.96	14.53	23.19
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	19.71	24.81	20.64	27.13	20.87	14.14	12.28	19.54	21.79	18.8
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	15.18	15.83	21.27	22.8	15.22	21.51	21.77	15.98	19.98
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	14.96	26.55	14.83	8.52	13.49	16.46	21.98	22.07
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	19.21	13.56	17.06	21.08	16.12	8.56	12.22
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	19.77	18.3	14.53	18.86	22.96	36.7
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	16.38	42.6	52.67	13.4	16.6
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	12.8	10.28	10.24	16.47
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	47.86	6.4	12.37
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	13.53	16.97
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	18.59
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Figure C.4: *PSkJW* matrix subsection of “Birds” documents (*corpus* II).



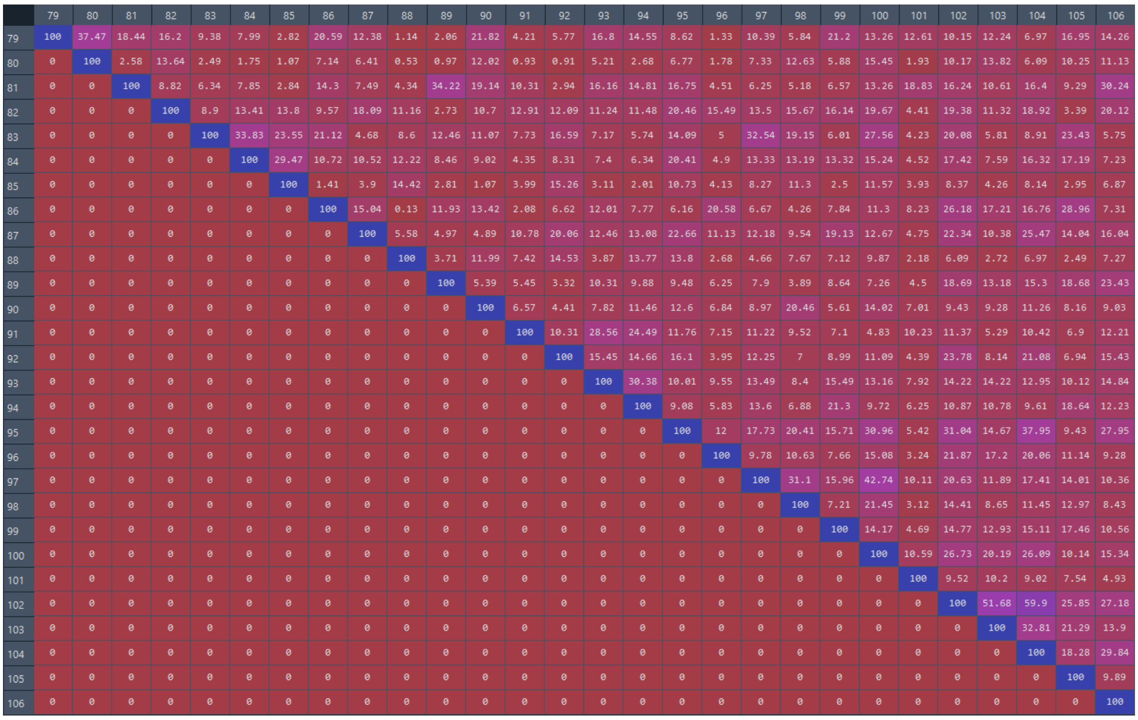
	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57
25	100	59.23	23.62	17.25	14.34	10.17	8.21	27.57	20.39	18.01	23.64	23.32	14.88	21.86	21.27	13.19	10.41	5.5	28.35	35.52	24.13	10.75	19.3	10.67	18.39	22.3	25.66	21.56	18.57	11.34	18.65	9.61	4.39
26	0	100	18.89	17.31	10.1	13.8	11.17	13.52	19.06	6.93	18.65	25.73	3.45	21.75	9.28	5.41	8.86	8.37	15.85	28.6	35.33	9.48	19.86	7.43	19.18	17.1	19.62	18.09	14.39	10.2	13.67	9.49	0
27	0	0	100	8.92	16.63	15.63	10.86	22.22	26.24	11.24	31.52	10.08	4.5	27.71	14.49	6.58	10.32	5.47	12.96	15.97	17.15	18.79	15.16	8.48	12.16	11.32	22.25	18.88	12.55	13.05	13.89	5.46	5.69
28	0	0	0	100	6.38	7	19.53	8.69	14.74	3.28	16.47	15.2	3.26	15.76	6.66	14.4	13.08	4.51	18.73	13.92	18.5	11.3	8.83	7.09	5.72	10.64	12.43	23.11	16.97	0.78	11.34	4.84	11.94
29	0	0	0	0	100	30.9	10.74	20	12.8	10.76	22.42	21.9	5.87	21.47	17.46	14.58	4.3	2.91	17.88	8.81	16.75	23.73	26.58	13.25	9.35	9.83	15.24	33.43	13.26	9.35	13.58	4.88	0.81
30	0	0	0	0	0	100	12.03	20.02	13.27	18.75	21.2	13.51	7.79	26.4	18.89	9.53	5.92	8.49	14.62	38.95	23.92	17.21	54.47	11.27	15.84	15.65	17.89	15.08	15.86	21.12	19.73	3.36	0.56
31	0	0	0	0	0	0	100	11.88	6.85	5.02	7.9	13.23	1.61	7.04	5.84	11.92	5.36	3.68	16.87	8.15	12.81	13.25	7.6	8.81	7.06	7.12	11.41	9.54	11.59	9.31	5.96	3.36	0.18
32	0	0	0	0	0	0	0	100	25.92	11.03	26.01	12.06	4.06	27.59	7.69	6.07	11.01	4.42	9.76	39.39	14.56	11.32	15.07	7.15	12.29	24.81	20.71	17.06	13.91	22.02	13.5	3.94	6.02
33	0	0	0	0	0	0	0	0	100	10.68	18.01	10.17	6.49	18.34	8.45	6.11	8.95	9	13.67	21.08	26.02	16.35	20.2	9.42	11.31	22.38	17.48	9.24	11.95	8.69	20.56	3.76	9.93
34	0	0	0	0	0	0	0	0	0	100	34.48	22.55	74.3	40.65	24.75	6.32	9.11	3.95	9.8	13.87	6.35	11	17.03	8.56	8.79	5.11	10.22	15.01	19.73	9.07	14.45	5.18	0.94
35	0	0	0	0	0	0	0	0	0	0	100	18.06	26.51	90.51	14.67	9.23	29.08	6.67	16.67	26.77	15.34	10.98	19.75	8.81	9.99	13.67	16.59	19.76	16.45	10.27	18.79	7.91	8.04
36	0	0	0	0	0	0	0	0	0	0	0	100	26.15	18.64	12.57	13.66	14.5	9.11	15.58	15.92	52.16	14.95	10.39	10.2	15.28	14.64	15.14	35.47	16.69	7.65	19.52	7	1.66
37	0	0	0	0	0	0	0	0	0	0	0	0	100	35.36	19.34	4.03	13.68	1.74	7	6	10.21	5.68	10.81	8.66	3.49	8.48	3.94	9.58	15.87	6.72	15.99	1.31	0
38	0	0	0	0	0	0	0	0	0	0	0	0	0	100	18.48	7.85	32.32	6.8	14.31	36.65	20.77	10.89	25.21	9.13	10.86	21.78	16.98	21.1	19.02	14.37	22.35	7.52	5.99
39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	46.26	7.51	7.89	29.6	13.34	9.37	9.92	17.44	8.81	15.86	11.49	11.34	13.2	13	5.66	8.68	4.33	0
40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	8.3	9.9	38.55	5.67	8.84	16.12	1.58	11.98	9.65	13.41	9.38	16.95	15.14	3.85	12.68	11.06	0
41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	6.44	11.89	10.57	10.58	8.95	11.65	7.5	4.22	14.6	9.7	10.05	15.4	7.47	18.54	9.83	2.72
42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	7.39	4.95	8.52	6.93	5.39	5.11	3.96	7.07	10.77	11.16	12.71	3.68	7.13	5.01	1.32
43	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	8.13	22.29	17.2	7.87	10.88	15.09	12.29	12.39	20.28	17.62	8.16	20.01	14.92	3.55
44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	18.05	12.89	32.46	5.7	20.19	39.38	17.82	11.6	14.61	24.35	12.64	5.94	4.44
45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	19.73	28.75	5.59	12.87	21.77	14.94	35.56	13	17.75	31.55	8.4	2.11
46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	17.18	15.49	12.21	12.29	10.9	23.35	12.36	21.91	13.66	24.95	1.61
47	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	7.64	15.05	38.1	13.05	9.36	12.36	19.63	29.83	5.87	3.67
48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	5.8	6.07	10.72	16.41	10.96	7.9	9.05	9.58	1.27
49	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	8.92	11.37	9.97	9.25	13.73	11.76	3.87	1.05
50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	14.81	12.61	13.67	11.5	15.71	3.64	1.2
51	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	19.74	22	8.91	13.74	8.42	4.44
52	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	19.83	6.35	13.27	7.89	2.52
53	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	6.51	13.68	8.38	2.19
54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	15.43	18.6	0.71

Figure C.5: *PSkJW* matrix subsection of “Cats” documents (*corpus* II).



	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78
57	100	18.88	24.14	8.44	13.11	4.68	13.3	9.01	21.76	12.27	19.14	18.15	11	19.82	23.89	21.31	23.29	21.73	3.94	30.48	25.26	20.77
58	0	100	15.6	8.18	16.85	5.67	13.99	27.89	19.62	11.06	19.09	21.49	7.18	19.62	30.1	24.77	22.77	18.18	8.73	37.61	22.33	21.98
59	0	0	100	22.16	14.62	8.75	15.18	12.45	10.56	24.99	18.98	34.53	11.16	17.91	23.44	18.33	24.04	14.41	25.07	25.86	21.72	19.7
60	0	0	0	100	6.27	3.13	9.9	11.73	11.62	30.43	20.82	13.92	18.37	14.43	6.18	7.63	9.98	4.88	33.39	13.98	11.51	4.04
61	0	0	0	0	100	10.8	14.69	10.38	19.9	6.62	22.94	18.96	7.74	17.65	26.72	27.7	19.18	22.64	11.11	23.39	19.88	23.35
62	0	0	0	0	0	100	27.01	4.08	8.54	5.86	10.09	1.32	6.91	1.91	12.79	7.49	10.63	15.93	1.82	7.63	5.72	9.02
63	0	0	0	0	0	0	100	15.43	22.29	5.81	15.58	5.31	10.61	4.03	9.29	11.74	7.98	18.97	10.49	10.08	8.21	10.08
64	0	0	0	0	0	0	0	100	29.29	22.42	26.29	21.14	9.49	11.81	9.45	10.49	43.73	16.79	30.35	25.02	24.62	31.03
65	0	0	0	0	0	0	0	0	100	12.93	25.92	15.74	9.29	26.4	30.79	19.92	27.37	32.77	12.39	37.54	29.32	34.66
66	0	0	0	0	0	0	0	0	0	100	19.62	20.52	14.73	6.96	9.03	8.78	28.4	9.06	25.14	19.71	16.59	12.12
67	0	0	0	0	0	0	0	0	0	0	100	20.75	17.55	24.22	27.73	24.92	31.75	24.28	33.32	33.39	33.16	22.98
68	0	0	0	0	0	0	0	0	0	0	0	100	6.48	23.88	40.47	27.31	44.72	28.76	25.12	46.29	30.39	35.78
69	0	0	0	0	0	0	0	0	0	0	0	0	100	6.53	7.55	14.2	10.65	7.55	11.34	15.35	16.34	5.53
70	0	0	0	0	0	0	0	0	0	0	0	0	0	100	37.38	22.47	23.25	19.46	13.36	44.09	29.92	42.29
71	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	46.79	43.16	46.1	5.33	57.59	44.92	49.42
72	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	30.65	28.47	6.86	38.8	28.58	29.02
73	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	36.44	22.32	48.92	41.87	54.34
74	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	8.18	42.63	37.77	38.94
75	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	17.39	13.82	13.31
76	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	49.98	53.72
77	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	40.25
78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Figure C.6: *PSkJW* matrix subsection of “Fish” documents (*corpus* II).

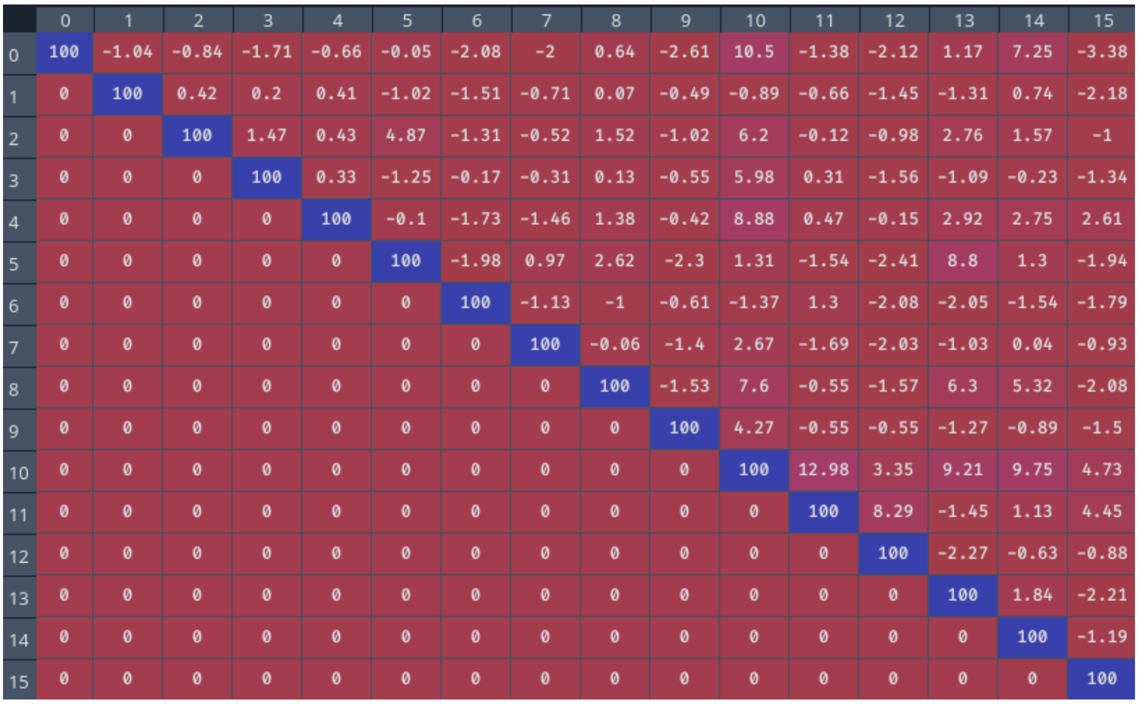


	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106
79	100	37.47	18.44	16.2	9.38	7.99	2.82	20.59	12.38	1.14	2.06	21.82	4.21	5.77	16.8	14.55	8.62	1.33	10.39	5.84	21.2	13.26	12.61	10.15	12.24	6.97	16.95	14.26
80	0	100	2.58	13.64	2.49	1.75	1.07	7.14	6.41	0.53	0.97	12.02	0.93	0.91	5.21	2.68	6.77	1.78	7.33	12.63	5.88	15.45	1.93	10.17	13.82	6.09	10.25	11.13
81	0	0	100	8.82	6.34	7.85	2.84	14.3	7.49	4.34	34.22	19.14	10.31	2.94	16.16	14.81	16.75	4.51	6.25	5.18	6.57	13.26	18.83	16.24	10.61	16.4	9.29	30.24
82	0	0	0	100	8.9	13.41	13.8	9.57	18.09	11.16	2.73	10.7	12.91	12.09	11.24	11.48	20.46	15.49	13.5	15.67	16.14	19.67	4.41	19.38	11.32	18.92	3.39	20.12
83	0	0	0	0	100	33.83	23.55	21.12	4.68	8.6	12.46	11.07	7.73	16.59	7.17	5.74	14.09	5	32.54	19.15	6.01	27.56	4.23	20.08	5.81	8.91	23.43	5.75
84	0	0	0	0	0	100	29.47	10.72	10.52	12.22	8.46	9.02	4.35	8.31	7.4	6.34	20.41	4.9	13.33	13.19	13.32	15.24	4.52	17.42	7.59	16.32	17.19	7.23
85	0	0	0	0	0	0	100	1.41	3.9	14.42	2.81	1.07	3.99	15.26	3.11	2.01	10.73	4.13	8.27	11.3	2.5	11.57	3.93	8.37	4.26	8.14	2.95	6.87
86	0	0	0	0	0	0	0	100	15.04	0.13	11.93	13.42	2.08	6.62	12.01	7.77	6.16	20.58	6.67	4.26	7.84	11.3	8.23	26.18	17.21	16.76	28.96	7.31
87	0	0	0	0	0	0	0	0	100	5.58	4.97	4.89	10.78	20.06	12.46	13.08	22.66	11.13	12.18	9.54	19.13	12.67	4.75	22.34	10.38	25.47	14.04	16.04
88	0	0	0	0	0	0	0	0	0	100	3.71	11.99	7.42	14.53	3.87	13.77	13.8	2.68	4.66	7.67	7.12	9.87	2.18	6.09	2.72	6.97	2.49	7.27
89	0	0	0	0	0	0	0	0	0	0	100	5.39	5.45	3.32	10.31	9.88	9.48	6.25	7.9	3.89	8.64	7.26	4.5	18.69	13.18	15.3	18.68	23.43
90	0	0	0	0	0	0	0	0	0	0	0	100	6.57	4.41	7.82	11.46	12.6	6.84	8.97	20.46	5.61	14.02	7.01	9.43	9.28	11.26	8.16	9.03
91	0	0	0	0	0	0	0	0	0	0	0	0	100	10.31	28.56	24.49	11.76	7.15	11.22	9.52	7.1	4.83	10.23	11.37	5.29	10.42	6.9	12.21
92	0	0	0	0	0	0	0	0	0	0	0	0	0	100	15.45	14.66	16.1	3.95	12.25	7	8.99	11.09	4.39	23.78	8.14	21.08	6.94	15.43
93	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	30.38	10.01	9.55	13.49	8.4	15.49	13.16	7.92	14.22	14.22	12.95	10.12	14.84
94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	9.08	5.83	13.6	6.88	21.3	9.72	6.25	10.87	10.78	9.61	18.64	12.23
95	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	12	17.73	20.41	15.71	30.96	5.42	31.04	14.67	37.95	9.43	27.95
96	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	9.78	10.63	7.66	15.08	3.24	21.87	17.2	20.06	11.14	9.28
97	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	31.1	15.96	42.74	10.11	20.63	11.89	17.41	14.01	10.36
98	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	7.21	21.45	3.12	14.41	8.65	11.45	12.97	8.43
99	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	14.17	4.69	14.77	12.93	15.11	17.46	10.56
100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	10.59	26.73	20.19	26.09	10.14	15.34
101	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	9.52	10.2	9.02	7.54	4.93
102	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	51.68	59.9	25.85	27.18
103	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	32.81	21.29	13.9
104	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	18.28	29.84
105	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	9.89
106	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Figure C.7: *PSkJW* matrix subsection of “Dogs” documents (*corpus* II).

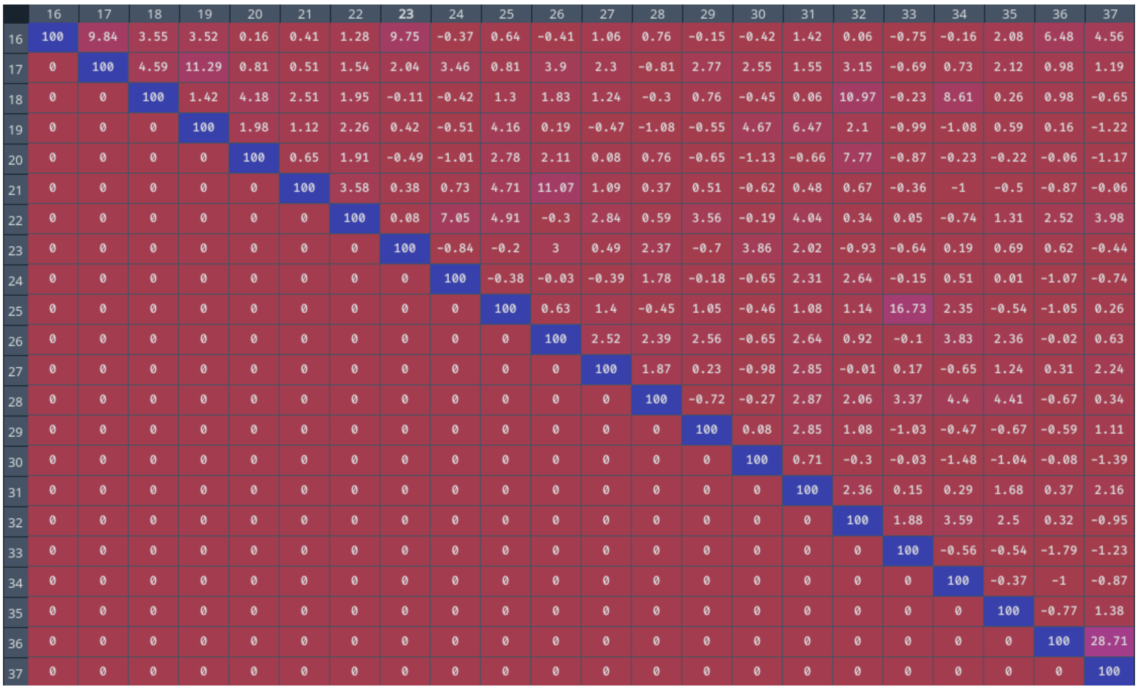
D

4th and 5th Moment matrices



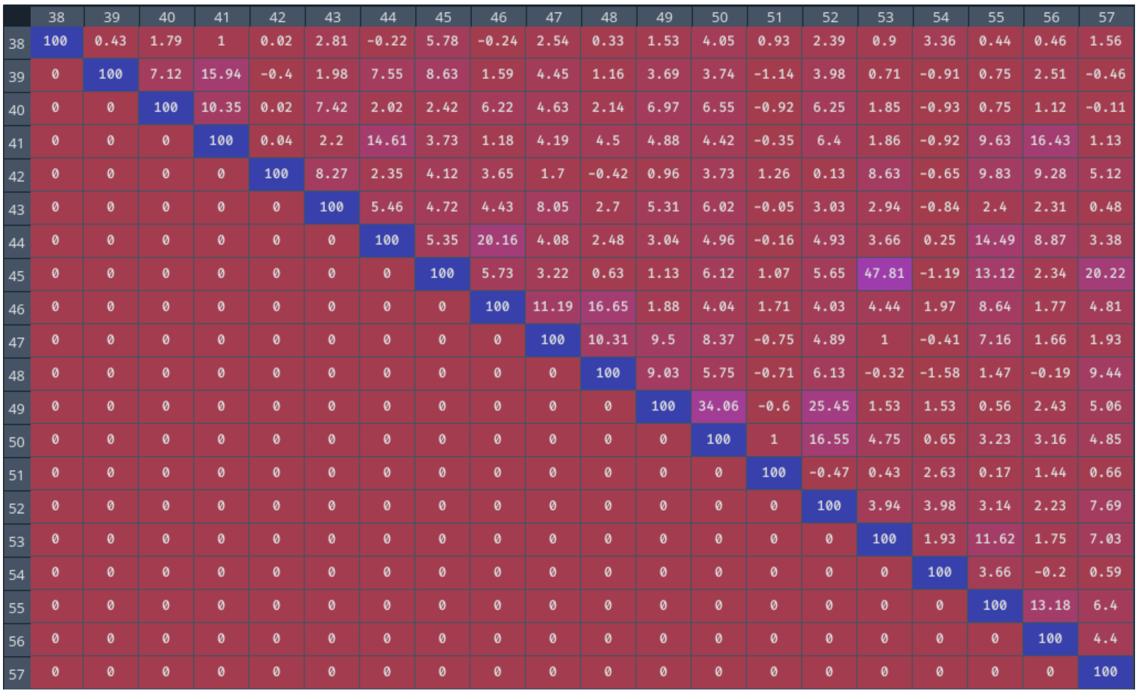
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	100	-1.04	-0.84	-1.71	-0.66	-0.05	-2.08	-2	0.64	-2.61	10.5	-1.38	-2.12	1.17	7.25	-3.38
1	0	100	0.42	0.2	0.41	-1.02	-1.51	-0.71	0.07	-0.49	-0.89	-0.66	-1.45	-1.31	0.74	-2.18
2	0	0	100	1.47	0.43	4.87	-1.31	-0.52	1.52	-1.02	6.2	-0.12	-0.98	2.76	1.57	-1
3	0	0	0	100	0.33	-1.25	-0.17	-0.31	0.13	-0.55	5.98	0.31	-1.56	-1.09	-0.23	-1.34
4	0	0	0	0	100	-0.1	-1.73	-1.46	1.38	-0.42	8.88	0.47	-0.15	2.92	2.75	2.61
5	0	0	0	0	0	100	-1.98	0.97	2.62	-2.3	1.31	-1.54	-2.41	8.8	1.3	-1.94
6	0	0	0	0	0	0	100	-1.13	-1	-0.61	-1.37	1.3	-2.08	-2.05	-1.54	-1.79
7	0	0	0	0	0	0	0	100	-0.06	-1.4	2.67	-1.69	-2.03	-1.03	0.04	-0.93
8	0	0	0	0	0	0	0	0	100	-1.53	7.6	-0.55	-1.57	6.3	5.32	-2.08
9	0	0	0	0	0	0	0	0	0	100	4.27	-0.55	-0.55	-1.27	-0.89	-1.5
10	0	0	0	0	0	0	0	0	0	0	100	12.98	3.35	9.21	9.75	4.73
11	0	0	0	0	0	0	0	0	0	0	0	100	8.29	-1.45	1.13	4.45
12	0	0	0	0	0	0	0	0	0	0	0	0	100	-2.27	-0.63	-0.88
13	0	0	0	0	0	0	0	0	0	0	0	0	0	100	1.84	-2.21
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	-1.19
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Figure D.1: 4th Moment matrix subsection of "Chemicals" documents (*corpus* I).



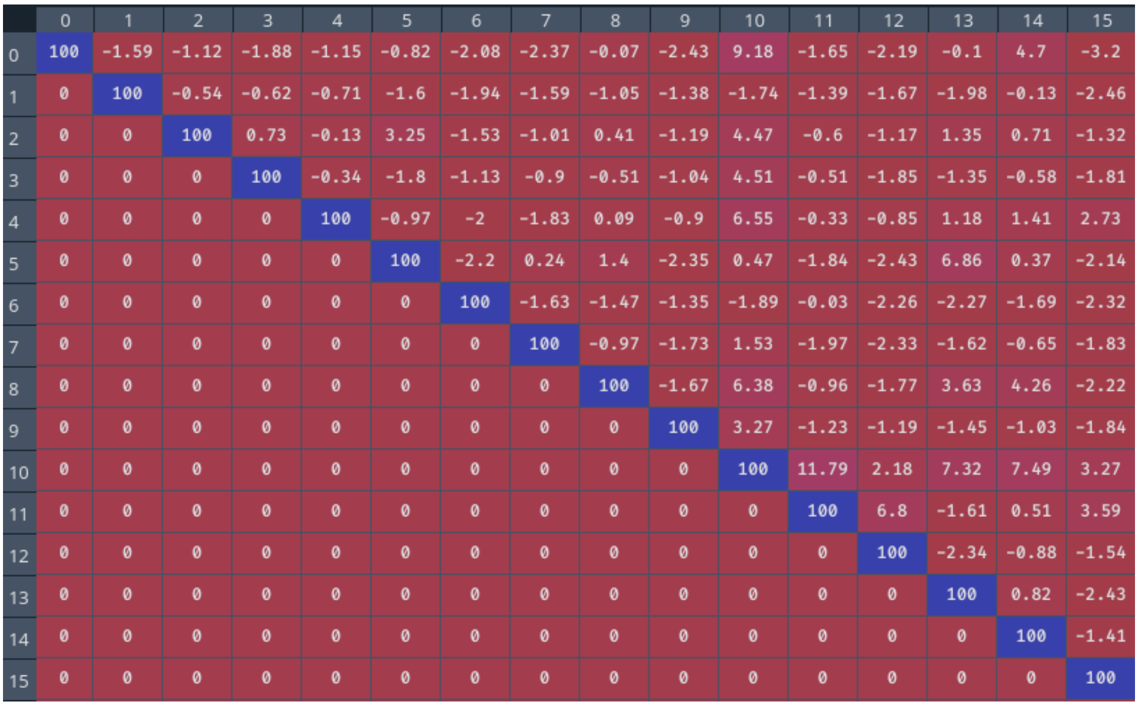
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
16	100	9.84	3.55	3.52	0.16	0.41	1.28	9.75	-0.37	0.64	-0.41	1.06	0.76	-0.15	-0.42	1.42	0.06	-0.75	-0.16	2.08	6.48	4.56
17	0	100	4.59	11.29	0.81	0.51	1.54	2.04	3.46	0.81	3.9	2.3	-0.81	2.77	2.55	1.55	3.15	-0.69	0.73	2.12	0.98	1.19
18	0	0	100	1.42	4.18	2.51	1.95	-0.11	-0.42	1.3	1.83	1.24	-0.3	0.76	-0.45	0.06	10.97	-0.23	8.61	0.26	0.98	-0.65
19	0	0	0	100	1.98	1.12	2.26	0.42	-0.51	4.16	0.19	-0.47	-1.08	-0.55	4.67	6.47	2.1	-0.99	-1.08	0.59	0.16	-1.22
20	0	0	0	0	100	0.65	1.91	-0.49	-1.01	2.78	2.11	0.08	0.76	-0.65	-1.13	-0.66	7.77	-0.87	-0.23	-0.22	-0.06	-1.17
21	0	0	0	0	0	100	3.58	0.38	0.73	4.71	11.07	1.09	0.37	0.51	-0.62	0.48	0.67	-0.36	-1	-0.5	-0.87	-0.06
22	0	0	0	0	0	0	100	0.08	7.05	4.91	-0.3	2.84	0.59	3.56	-0.19	4.04	0.34	0.05	-0.74	1.31	2.52	3.98
23	0	0	0	0	0	0	0	100	-0.84	-0.2	3	0.49	2.37	-0.7	3.86	2.02	-0.93	-0.64	0.19	0.69	0.62	-0.44
24	0	0	0	0	0	0	0	0	100	-0.38	-0.03	-0.39	1.78	-0.18	-0.65	2.31	2.64	-0.15	0.51	0.01	-1.07	-0.74
25	0	0	0	0	0	0	0	0	0	100	0.63	1.4	-0.45	1.05	-0.46	1.08	1.14	16.73	2.35	-0.54	-1.05	0.26
26	0	0	0	0	0	0	0	0	0	0	100	2.52	2.39	2.56	-0.65	2.64	0.92	-0.1	3.83	2.36	-0.02	0.63
27	0	0	0	0	0	0	0	0	0	0	0	100	1.87	0.23	-0.98	2.85	-0.01	0.17	-0.65	1.24	0.31	2.24
28	0	0	0	0	0	0	0	0	0	0	0	0	100	-0.72	-0.27	2.87	2.06	3.37	4.4	4.41	-0.67	0.34
29	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0.08	2.85	1.08	-1.03	-0.47	-0.67	-0.59	1.11
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0.71	-0.3	-0.03	-1.48	-1.04	-0.08	-1.39
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	2.36	0.15	0.29	1.68	0.37	2.16
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	1.88	3.59	2.5	0.32	-0.95
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	-0.56	-0.54	-1.79	-1.23
34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	-0.37	-1	-0.87
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	-0.77	1.38
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	28.71
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Figure D.2: 4th Moment matrix subsection of "Constellations" documents (*corpus* I).



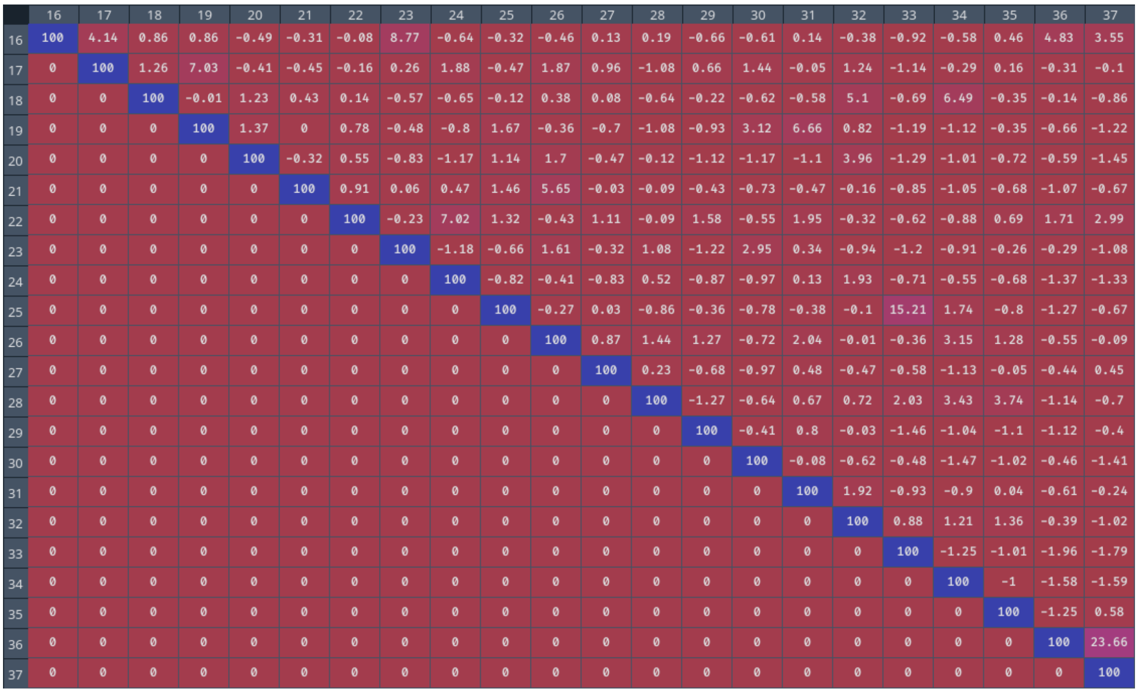
	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57
38	100	0.43	1.79	1	0.02	2.81	-0.22	5.78	-0.24	2.54	0.33	1.53	4.05	0.93	2.39	0.9	3.36	0.44	0.46	1.56
39	0	100	7.12	15.94	-0.4	1.98	7.55	8.63	1.59	4.45	1.16	3.69	3.74	-1.14	3.98	0.71	-0.91	0.75	2.51	-0.46
40	0	0	100	10.35	0.02	7.42	2.02	2.42	6.22	4.63	2.14	6.97	6.55	-0.92	6.25	1.85	-0.93	0.75	1.12	-0.11
41	0	0	0	100	0.04	2.2	14.61	3.73	1.18	4.19	4.5	4.88	4.42	-0.35	6.4	1.86	-0.92	9.63	16.43	1.13
42	0	0	0	0	100	8.27	2.35	4.12	3.65	1.7	-0.42	0.96	3.73	1.26	0.13	8.63	-0.65	9.83	9.28	5.12
43	0	0	0	0	0	100	5.46	4.72	4.43	8.05	2.7	5.31	6.02	-0.05	3.03	2.94	-0.84	2.4	2.31	0.48
44	0	0	0	0	0	0	100	5.35	20.16	4.08	2.48	3.04	4.96	-0.16	4.93	3.66	0.25	14.49	8.87	3.38
45	0	0	0	0	0	0	0	100	5.73	3.22	0.63	1.13	6.12	1.07	5.65	47.81	-1.19	13.12	2.34	20.22
46	0	0	0	0	0	0	0	0	100	11.19	16.65	1.88	4.04	1.71	4.03	4.44	1.97	8.64	1.77	4.81
47	0	0	0	0	0	0	0	0	0	100	10.31	9.5	8.37	-0.75	4.89	1	-0.41	7.16	1.66	1.93
48	0	0	0	0	0	0	0	0	0	0	100	9.03	5.75	-0.71	6.13	-0.32	-1.58	1.47	-0.19	9.44
49	0	0	0	0	0	0	0	0	0	0	0	100	34.06	-0.6	25.45	1.53	1.53	0.56	2.43	5.06
50	0	0	0	0	0	0	0	0	0	0	0	0	100	1	16.55	4.75	0.65	3.23	3.16	4.85
51	0	0	0	0	0	0	0	0	0	0	0	0	0	100	-0.47	0.43	2.63	0.17	1.44	0.66
52	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	3.94	3.98	3.14	2.23	7.69
53	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	1.93	11.62	1.75	7.03
54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	3.66	-0.2	0.59
55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	13.18	6.4
56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	4.4
57	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Figure D.3: 4th Moment matrix subsection of "Tennis" documents (*corpus* I).



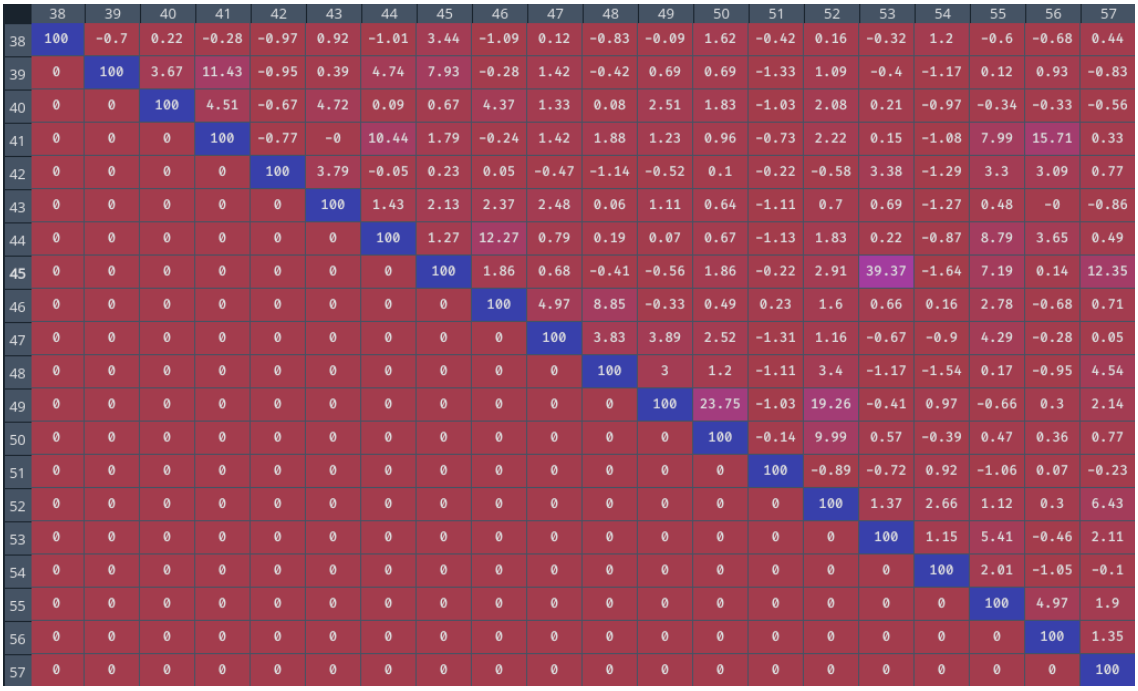
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	100	-1.59	-1.12	-1.88	-1.15	-0.82	-2.08	-2.37	-0.07	-2.43	9.18	-1.65	-2.19	-0.1	4.7	-3.2
1	0	100	-0.54	-0.62	-0.71	-1.6	-1.94	-1.59	-1.05	-1.38	-1.74	-1.39	-1.67	-1.98	-0.13	-2.46
2	0	0	100	0.73	-0.13	3.25	-1.53	-1.01	0.41	-1.19	4.47	-0.6	-1.17	1.35	0.71	-1.32
3	0	0	0	100	-0.34	-1.8	-1.13	-0.9	-0.51	-1.04	4.51	-0.51	-1.85	-1.35	-0.58	-1.81
4	0	0	0	0	100	-0.97	-2	-1.83	0.09	-0.9	6.55	-0.33	-0.85	1.18	1.41	2.73
5	0	0	0	0	0	100	-2.2	0.24	1.4	-2.35	0.47	-1.84	-2.43	6.86	0.37	-2.14
6	0	0	0	0	0	0	100	-1.63	-1.47	-1.35	-1.89	-0.03	-2.26	-2.27	-1.69	-2.32
7	0	0	0	0	0	0	0	100	-0.97	-1.73	1.53	-1.97	-2.33	-1.62	-0.65	-1.83
8	0	0	0	0	0	0	0	0	100	-1.67	6.38	-0.96	-1.77	3.63	4.26	-2.22
9	0	0	0	0	0	0	0	0	0	100	3.27	-1.23	-1.19	-1.45	-1.03	-1.84
10	0	0	0	0	0	0	0	0	0	0	100	11.79	2.18	7.32	7.49	3.27
11	0	0	0	0	0	0	0	0	0	0	0	100	6.8	-1.61	0.51	3.59
12	0	0	0	0	0	0	0	0	0	0	0	0	100	-2.34	-0.88	-1.54
13	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0.82	-2.43
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	-1.41
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Figure D.4: 5th Moment matrix subsection of "Chemicals" documents (*corpus* I).



	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
16	100	4.14	0.86	0.86	-0.49	-0.31	-0.08	8.77	-0.64	-0.32	-0.46	0.13	0.19	-0.66	-0.61	0.14	-0.38	-0.92	-0.58	0.46	4.83	3.55
17	0	100	1.26	7.03	-0.41	-0.45	-0.16	0.26	1.88	-0.47	1.87	0.96	-1.08	0.66	1.44	-0.05	1.24	-1.14	-0.29	0.16	-0.31	-0.1
18	0	0	100	-0.01	1.23	0.43	0.14	-0.57	-0.65	-0.12	0.38	0.08	-0.64	-0.22	-0.62	-0.58	5.1	-0.69	6.49	-0.35	-0.14	-0.86
19	0	0	0	100	1.37	0	0.78	-0.48	-0.8	1.67	-0.36	-0.7	-1.08	-0.93	3.12	6.66	0.82	-1.19	-1.12	-0.35	-0.66	-1.22
20	0	0	0	0	100	-0.32	0.55	-0.83	-1.17	1.14	1.7	-0.47	-0.12	-1.12	-1.17	-1.1	3.96	-1.29	-1.01	-0.72	-0.59	-1.45
21	0	0	0	0	0	100	0.91	0.06	0.47	1.46	5.65	-0.03	-0.09	-0.43	-0.73	-0.47	-0.16	-0.85	-1.05	-0.68	-1.07	-0.67
22	0	0	0	0	0	0	100	-0.23	7.02	1.32	-0.43	1.11	-0.09	1.58	-0.55	1.95	-0.32	-0.62	-0.88	0.69	1.71	2.99
23	0	0	0	0	0	0	0	100	-1.18	-0.66	1.61	-0.32	1.08	-1.22	2.95	0.34	-0.94	-1.2	-0.91	-0.26	-0.29	-1.08
24	0	0	0	0	0	0	0	0	100	-0.82	-0.41	-0.83	0.52	-0.87	-0.97	0.13	1.93	-0.71	-0.55	-0.68	-1.37	-1.33
25	0	0	0	0	0	0	0	0	0	100	-0.27	0.03	-0.86	-0.36	-0.78	-0.38	-0.1	15.21	1.74	-0.8	-1.27	-0.67
26	0	0	0	0	0	0	0	0	0	0	100	0.87	1.44	1.27	-0.72	2.04	-0.01	-0.36	3.15	1.28	-0.55	-0.09
27	0	0	0	0	0	0	0	0	0	0	0	100	0.23	-0.68	-0.97	0.48	-0.47	-0.58	-1.13	-0.05	-0.44	0.45
28	0	0	0	0	0	0	0	0	0	0	0	0	100	-1.27	-0.64	0.67	0.72	2.03	3.43	3.74	-1.14	-0.7
29	0	0	0	0	0	0	0	0	0	0	0	0	0	100	-0.41	0.8	-0.03	-1.46	-1.04	-1.1	-1.12	-0.4
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	-0.08	-0.62	-0.48	-1.47	-1.02	-0.46	-1.41
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	1.92	-0.93	-0.9	0.04	-0.61	-0.24
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0.88	1.21	1.36	-0.39	-1.02
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	-1.25	-1.01	-1.96	-1.79
34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	-1	-1.58	-1.59
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	-1.25	0.58
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	23.66
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Figure D.5: 5th Moment matrix subsection of "Constellations" documents (*corpus* I).

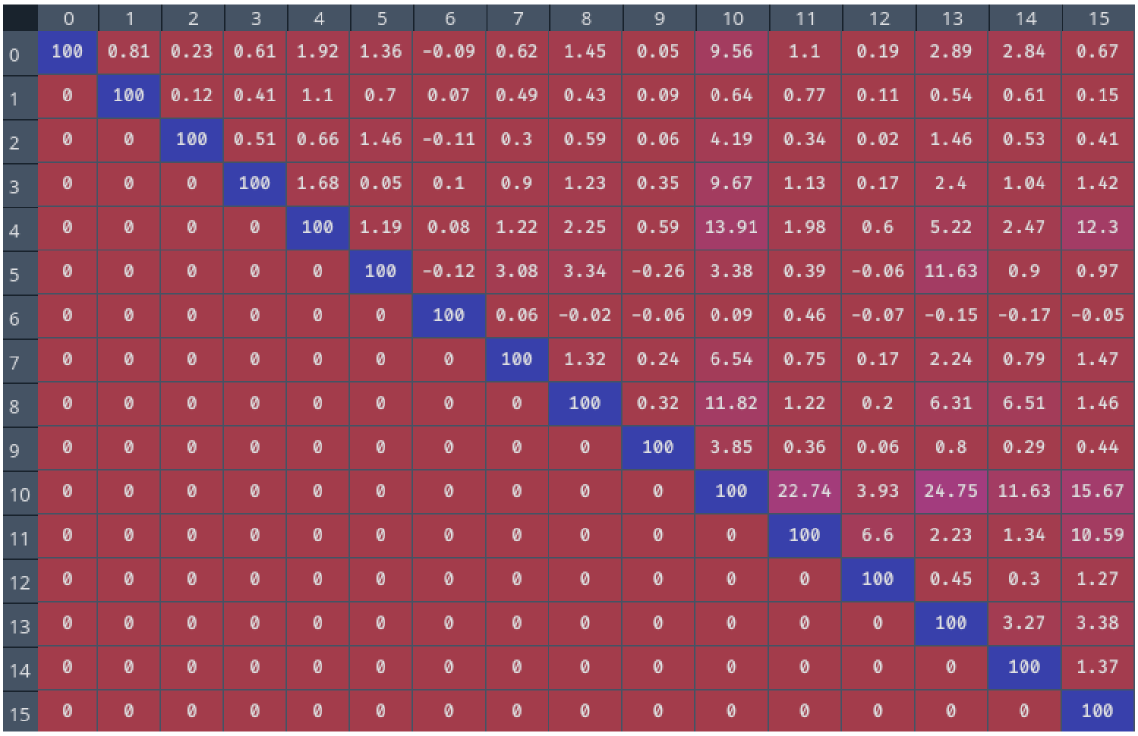


	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57
38	100	-0.7	0.22	-0.28	-0.97	0.92	-1.01	3.44	-1.09	0.12	-0.83	-0.09	1.62	-0.42	0.16	-0.32	1.2	-0.6	-0.68	0.44
39	0	100	3.67	11.43	-0.95	0.39	4.74	7.93	-0.28	1.42	-0.42	0.69	0.69	-1.33	1.09	-0.4	-1.17	0.12	0.93	-0.83
40	0	0	100	4.51	-0.67	4.72	0.09	0.67	4.37	1.33	0.08	2.51	1.83	-1.03	2.08	0.21	-0.97	-0.34	-0.33	-0.56
41	0	0	0	100	-0.77	-0	10.44	1.79	-0.24	1.42	1.88	1.23	0.96	-0.73	2.22	0.15	-1.08	7.99	15.71	0.33
42	0	0	0	0	100	3.79	-0.05	0.23	0.05	-0.47	-1.14	-0.52	0.1	-0.22	-0.58	3.38	-1.29	3.3	3.09	0.77
43	0	0	0	0	0	100	1.43	2.13	2.37	2.48	0.06	1.11	0.64	-1.11	0.7	0.69	-1.27	0.48	-0	-0.86
44	0	0	0	0	0	0	100	1.27	12.27	0.79	0.19	0.07	0.67	-1.13	1.83	0.22	-0.87	8.79	3.65	0.49
45	0	0	0	0	0	0	0	100	1.86	0.68	-0.41	-0.56	1.86	-0.22	2.91	39.37	-1.64	7.19	0.14	12.35
46	0	0	0	0	0	0	0	0	100	4.97	8.85	-0.33	0.49	0.23	1.6	0.66	0.16	2.78	-0.68	0.71
47	0	0	0	0	0	0	0	0	0	100	3.83	3.89	2.52	-1.31	1.16	-0.67	-0.9	4.29	-0.28	0.05
48	0	0	0	0	0	0	0	0	0	0	100	3	1.2	-1.11	3.4	-1.17	-1.54	0.17	-0.95	4.54
49	0	0	0	0	0	0	0	0	0	0	0	100	23.75	-1.03	19.26	-0.41	0.97	-0.66	0.3	2.14
50	0	0	0	0	0	0	0	0	0	0	0	0	100	-0.14	9.99	0.57	-0.39	0.47	0.36	0.77
51	0	0	0	0	0	0	0	0	0	0	0	0	0	100	-0.89	-0.72	0.92	-1.06	0.07	-0.23
52	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	1.37	2.66	1.12	0.3	6.43
53	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	1.15	5.41	-0.46	2.11
54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	2.01	-1.05	-0.1
55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	4.97	1.9
56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	1.35
57	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Figure D.6: 5th Moment matrix subsection of “Tennis” documents (*corpus* I).

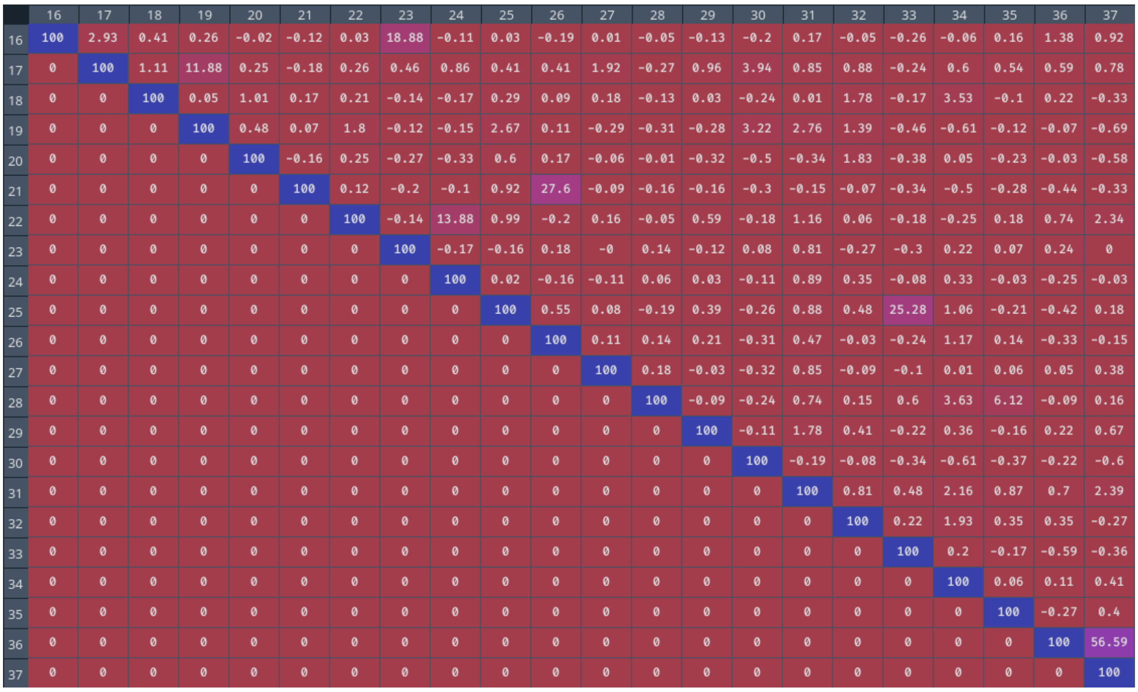
E

Probability Jump matrices



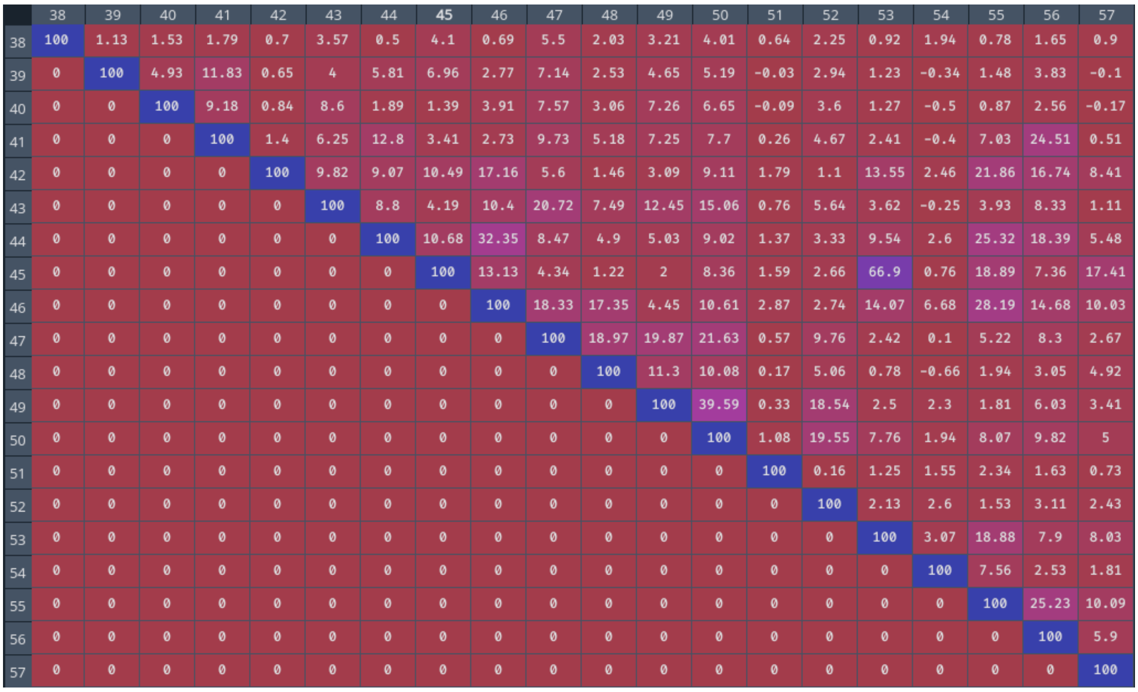
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	100	0.81	0.23	0.61	1.92	1.36	-0.09	0.62	1.45	0.05	9.56	1.1	0.19	2.89	2.84	0.67
1	0	100	0.12	0.41	1.1	0.7	0.07	0.49	0.43	0.09	0.64	0.77	0.11	0.54	0.61	0.15
2	0	0	100	0.51	0.66	1.46	-0.11	0.3	0.59	0.06	4.19	0.34	0.02	1.46	0.53	0.41
3	0	0	0	100	1.68	0.05	0.1	0.9	1.23	0.35	9.67	1.13	0.17	2.4	1.04	1.42
4	0	0	0	0	100	1.19	0.08	1.22	2.25	0.59	13.91	1.98	0.6	5.22	2.47	12.3
5	0	0	0	0	0	100	-0.12	3.08	3.34	-0.26	3.38	0.39	-0.06	11.63	0.9	0.97
6	0	0	0	0	0	0	100	0.06	-0.02	-0.06	0.09	0.46	-0.07	-0.15	-0.17	-0.05
7	0	0	0	0	0	0	0	100	1.32	0.24	6.54	0.75	0.17	2.24	0.79	1.47
8	0	0	0	0	0	0	0	0	100	0.32	11.82	1.22	0.2	6.31	6.51	1.46
9	0	0	0	0	0	0	0	0	0	100	3.85	0.36	0.06	0.8	0.29	0.44
10	0	0	0	0	0	0	0	0	0	0	100	22.74	3.93	24.75	11.63	15.67
11	0	0	0	0	0	0	0	0	0	0	0	100	6.6	2.23	1.34	10.59
12	0	0	0	0	0	0	0	0	0	0	0	0	100	0.45	0.3	1.27
13	0	0	0	0	0	0	0	0	0	0	0	0	0	100	3.27	3.38
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	1.37
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Figure E.1: Jump matrix subsection of "Chemicals" documents (*corpus* I).



	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
16	100	2.93	0.41	0.26	-0.02	-0.12	0.03	18.88	-0.11	0.03	-0.19	0.01	-0.05	-0.13	-0.2	0.17	-0.05	-0.26	-0.06	0.16	1.38	0.92
17	0	100	1.11	11.88	0.25	-0.18	0.26	0.46	0.86	0.41	0.41	1.92	-0.27	0.96	3.94	0.85	0.88	-0.24	0.6	0.54	0.59	0.78
18	0	0	100	0.05	1.01	0.17	0.21	-0.14	-0.17	0.29	0.09	0.18	-0.13	0.03	-0.24	0.01	1.78	-0.17	3.53	-0.1	0.22	-0.33
19	0	0	0	100	0.48	0.07	1.8	-0.12	-0.15	2.67	0.11	-0.29	-0.31	-0.28	3.22	2.76	1.39	-0.46	-0.61	-0.12	-0.07	-0.69
20	0	0	0	0	100	-0.16	0.25	-0.27	-0.33	0.6	0.17	-0.06	-0.01	-0.32	-0.5	-0.34	1.83	-0.38	0.05	-0.23	-0.03	-0.58
21	0	0	0	0	0	100	0.12	-0.2	-0.1	0.92	27.6	-0.09	-0.16	-0.16	-0.3	-0.15	-0.07	-0.34	-0.5	-0.28	-0.44	-0.33
22	0	0	0	0	0	0	100	-0.14	13.88	0.99	-0.2	0.16	-0.05	0.59	-0.18	1.16	0.06	-0.18	-0.25	0.18	0.74	2.34
23	0	0	0	0	0	0	0	100	-0.17	-0.16	0.18	-0	0.14	-0.12	0.08	0.81	-0.27	-0.3	0.22	0.07	0.24	0
24	0	0	0	0	0	0	0	0	100	0.02	-0.16	-0.11	0.06	0.03	-0.11	0.89	0.35	-0.08	0.33	-0.03	-0.25	-0.03
25	0	0	0	0	0	0	0	0	0	100	0.55	0.08	-0.19	0.39	-0.26	0.88	0.48	25.28	1.06	-0.21	-0.42	0.18
26	0	0	0	0	0	0	0	0	0	0	100	0.11	0.14	0.21	-0.31	0.47	-0.03	-0.24	1.17	0.14	-0.33	-0.15
27	0	0	0	0	0	0	0	0	0	0	0	100	0.18	-0.03	-0.32	0.85	-0.09	-0.1	0.01	0.06	0.05	0.38
28	0	0	0	0	0	0	0	0	0	0	0	0	100	-0.09	-0.24	0.74	0.15	0.6	3.63	6.12	-0.09	0.16
29	0	0	0	0	0	0	0	0	0	0	0	0	0	100	-0.11	1.78	0.41	-0.22	0.36	-0.16	0.22	0.67
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	-0.19	-0.08	-0.34	-0.61	-0.37	-0.22	-0.6
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0.81	0.48	2.16	0.87	0.7	2.39
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0.22	1.93	0.35	0.35	-0.27
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0.2	-0.17	-0.59	-0.36
34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0.06	0.11	0.41
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	-0.27	0.4
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	56.59
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Figure E.2: Jump matrix subsection of "Constellations" documents (*corpus* I).



	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57
38	100	1.13	1.53	1.79	0.7	3.57	0.5	4.1	0.69	5.5	2.03	3.21	4.01	0.64	2.25	0.92	1.94	0.78	1.65	0.9
39	0	100	4.93	11.83	0.65	4	5.81	6.96	2.77	7.14	2.53	4.65	5.19	-0.03	2.94	1.23	-0.34	1.48	3.83	-0.1
40	0	0	100	9.18	0.84	8.6	1.89	1.39	3.91	7.57	3.06	7.26	6.65	-0.09	3.6	1.27	-0.5	0.87	2.56	-0.17
41	0	0	0	100	1.4	6.25	12.8	3.41	2.73	9.73	5.18	7.25	7.7	0.26	4.67	2.41	-0.4	7.03	24.51	0.51
42	0	0	0	0	100	9.82	9.07	10.49	17.16	5.6	1.46	3.09	9.11	1.79	1.1	13.55	2.46	21.86	16.74	8.41
43	0	0	0	0	0	100	8.8	4.19	10.4	20.72	7.49	12.45	15.06	0.76	5.64	3.62	-0.25	3.93	8.33	1.11
44	0	0	0	0	0	0	100	10.68	32.35	8.47	4.9	5.03	9.02	1.37	3.33	9.54	2.6	25.32	18.39	5.48
45	0	0	0	0	0	0	0	100	13.13	4.34	1.22	2	8.36	1.59	2.66	66.9	0.76	18.89	7.36	17.41
46	0	0	0	0	0	0	0	0	100	18.33	17.35	4.45	10.61	2.87	2.74	14.07	6.68	28.19	14.68	10.03
47	0	0	0	0	0	0	0	0	0	100	18.97	19.87	21.63	0.57	9.76	2.42	0.1	5.22	8.3	2.67
48	0	0	0	0	0	0	0	0	0	0	100	11.3	10.08	0.17	5.06	0.78	-0.66	1.94	3.05	4.92
49	0	0	0	0	0	0	0	0	0	0	0	100	39.59	0.33	18.54	2.5	2.3	1.81	6.03	3.41
50	0	0	0	0	0	0	0	0	0	0	0	0	100	1.08	19.55	7.76	1.94	8.07	9.82	5
51	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0.16	1.25	1.55	2.34	1.63	0.73
52	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	2.13	2.6	1.53	3.11	2.43
53	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	3.07	18.88	7.9	8.03
54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	7.56	2.53	1.81
55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	25.23	10.09
56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	5.9
57	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Figure E.3: Jump matrix subsection of "Tennis" documents (*corpus* I).

F

Document clustering

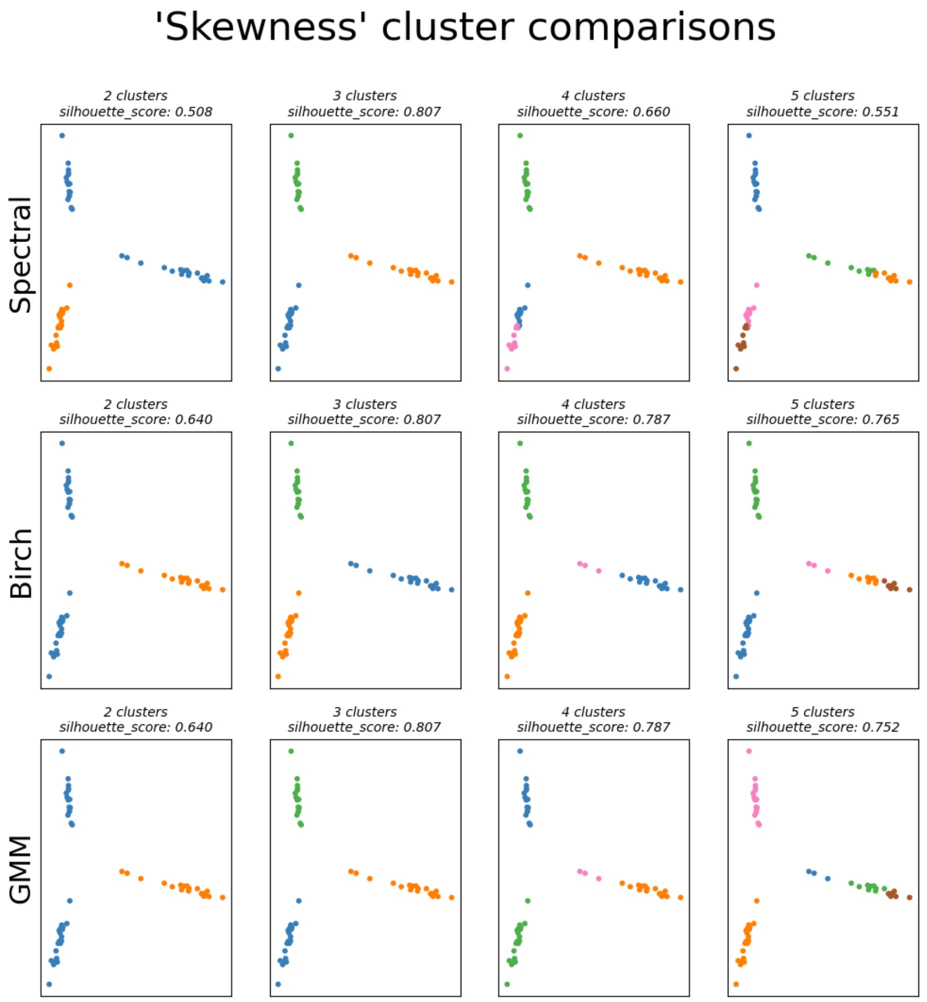


Figure F.1: Comparison of different clustering algorithms with Skewness metric and varying number of clusters (*corpus* I).

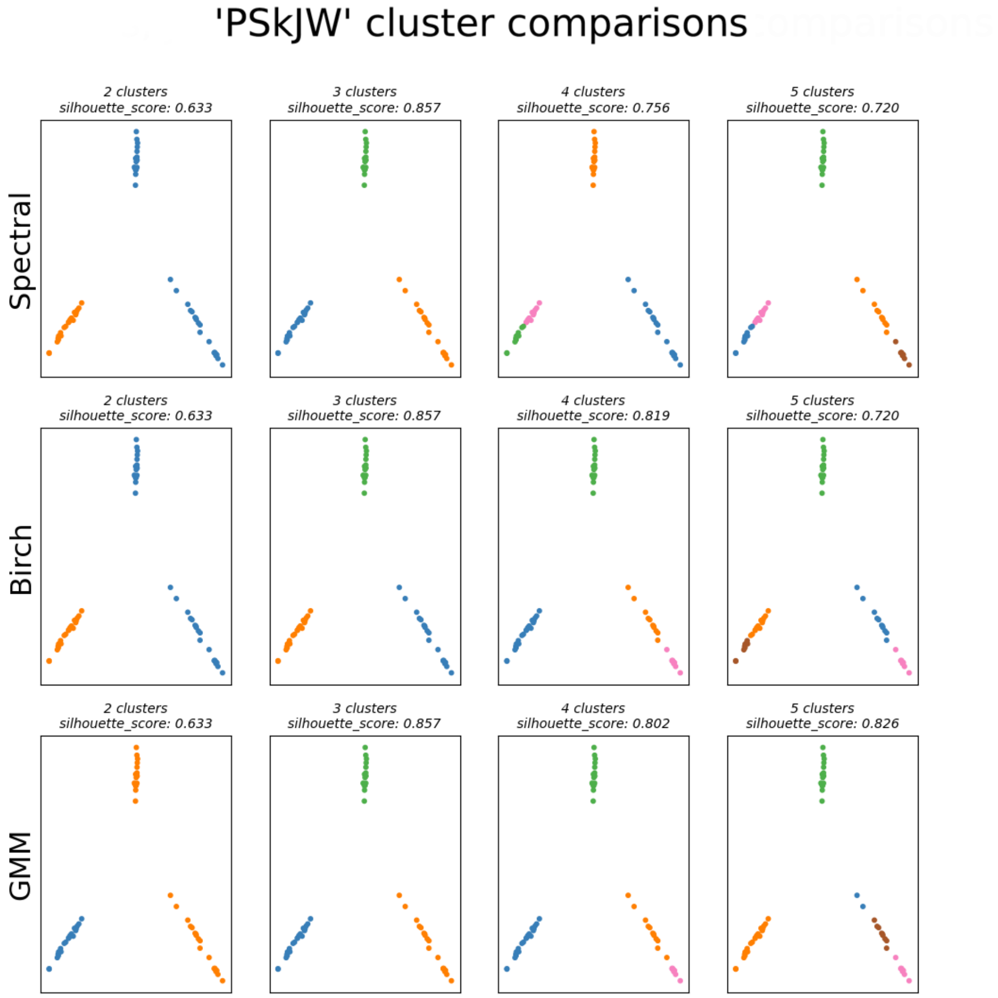


Figure F.2: Comparison of different clustering algorithms with $PSkJW$ metric and varying number of clusters (*corpus* I).

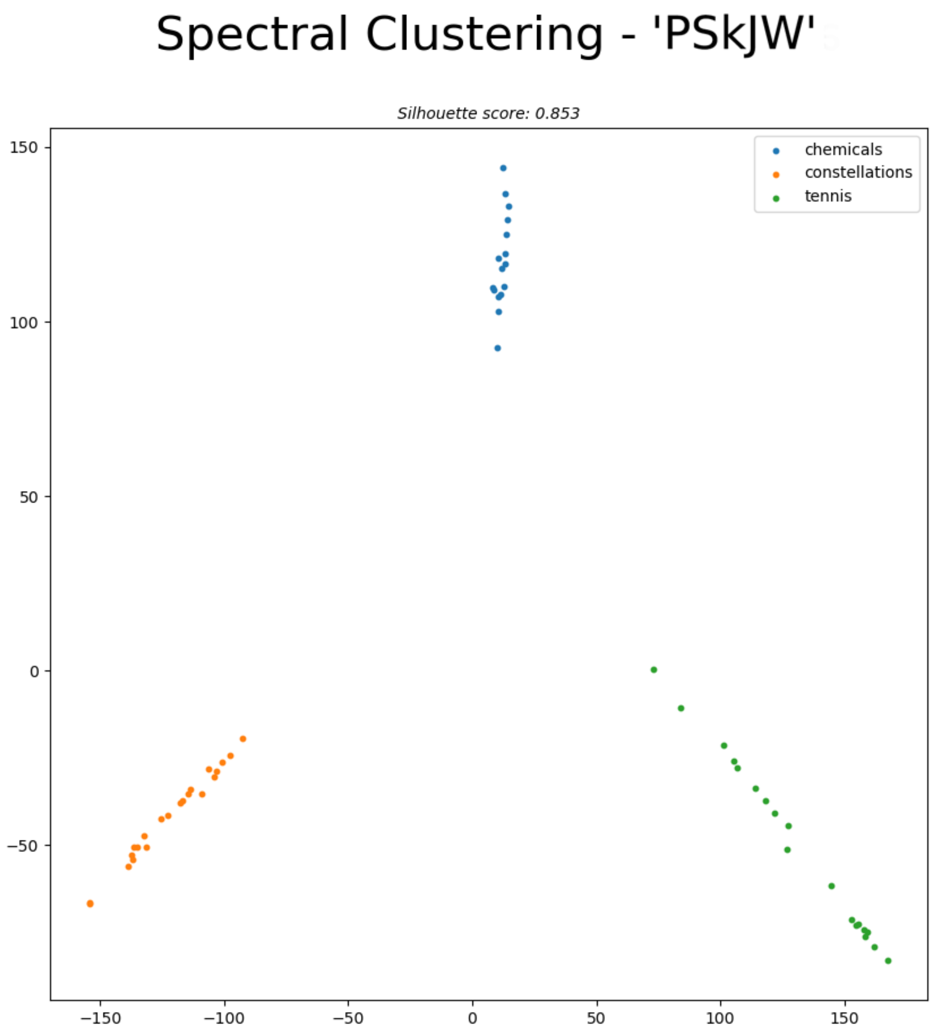


Figure F.3: Spectral Clustering results with *PSkJW* metric with 3 clusters (*corpus* I).

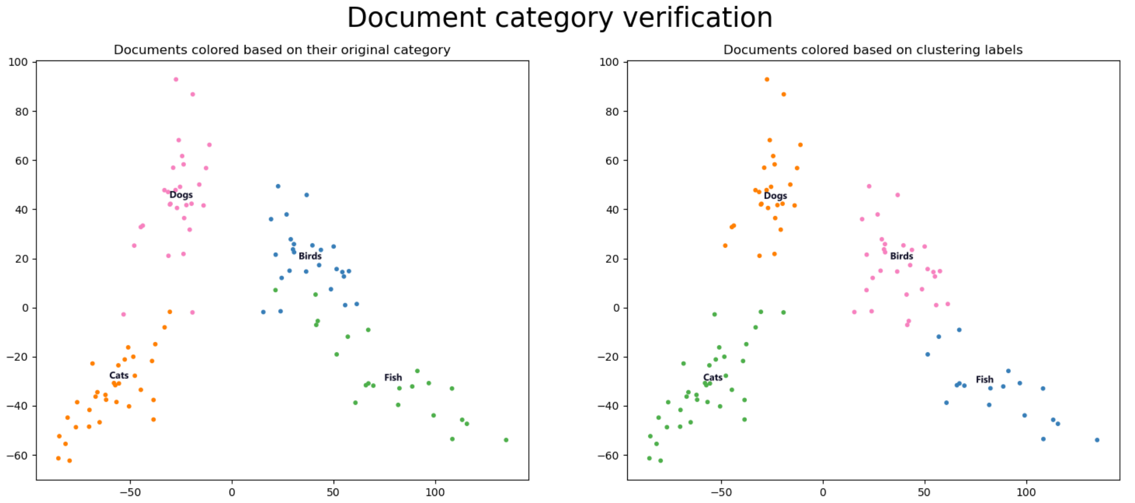


Figure F.4: Comparison between the original category of documents present in *corpus* II, and the predicted categories obtained by Spectral Clustering.

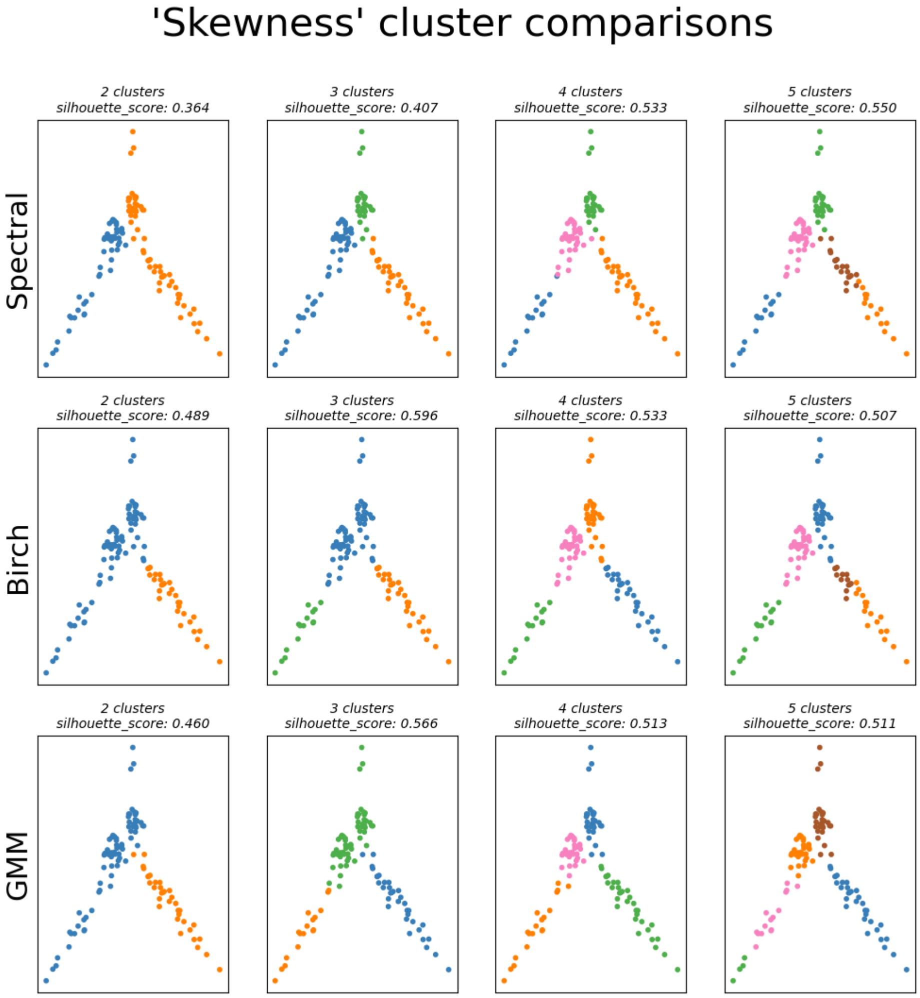


Figure F.5: Comparison of different clustering algorithms with Skewness metric and varying number of clusters (*corpus* II).

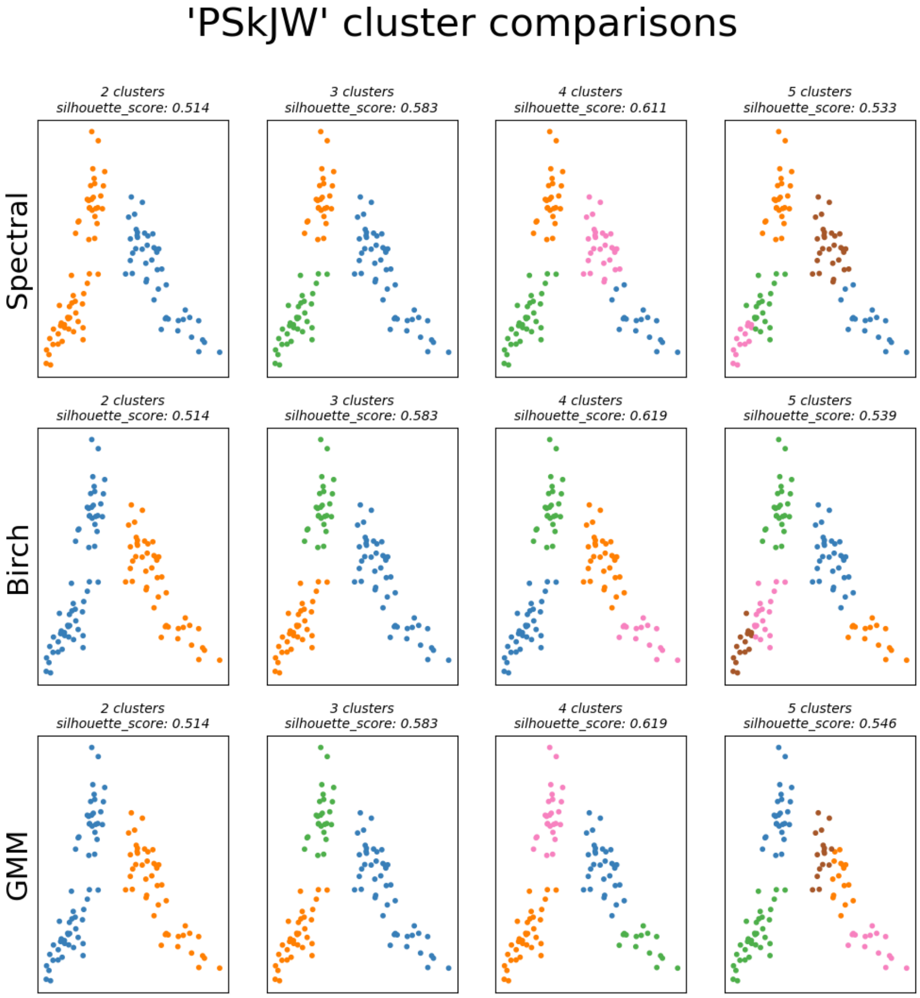


Figure F.6: Comparison of different clustering algorithms with $PSkJW$ metric and varying number of clusters (*corpus* II).

G

Classification similarities

Table G.1: Example of computed similarities between a test document ($Tennis_{50}$) and the training document set.

$Tennis_{50}$ similarities to documents in training set					
Files	Similarity	Files	Similarity	Files	Similarity
$chemicals_0$	-1.81	$constellations_{16}$	0.26	$tennis_{38}$	42.2
$chemicals_1$	-1.74	$constellations_{17}$	-1.01	$tennis_{39}$	56.6
$chemicals_2$	-1.58	$constellations_{18}$	-0.84	$tennis_{40}$	67.88
$chemicals_3$	-1.26	$constellations_{19}$	-1.03	$tennis_{41}$	58.78
$chemicals_4$	-1.08	$constellations_{20}$	-1.35	$tennis_{42}$	33.22
$chemicals_5$	-1.17	$constellations_{21}$	-1.05	$tennis_{43}$	60.75
$chemicals_6$	-1.85	$constellations_{22}$	-0.8	$tennis_{44}$	39.3
$chemicals_7$	-1.98	$constellations_{23}$	-1.73	$tennis_{45}$	35.94
$chemicals_8$	-0.87	$constellations_{24}$	-0.48	$tennis_{46}$	33.14
$chemicals_9$	-1.83	$constellations_{25}$	-0.92	$tennis_{47}$	66.48
$chemicals_{10}$	-0.97	$constellations_{26}$	-1.0	$tennis_{48}$	73.71
$chemicals_{11}$	-1.7	$constellations_{27}$	-1.22	$tennis_{49}$	87.69
$chemicals_{12}$	-2.24	$constellations_{28}$	-1.68	$tennis_{51}$	20.92
$chemicals_{13}$	-1.36	$constellations_{29}$	-1.35	$tennis_{52}$	67.12
$chemicals_{14}$	-1.27	$constellations_{30}$	-1.01	$tennis_{53}$	41.81
$chemicals_{15}$	-1.13	$constellations_{31}$	-0.89	$tennis_{54}$	15.06
		$constellations_{32}$	-0.91	$tennis_{55}$	23.97
		$constellations_{33}$	-1.8	$tennis_{56}$	32.97
		$constellations_{34}$	-1.57	$tennis_{57}$	41.92
		$constellations_{35}$	-1.27		
		$constellations_{36}$	-0.47		
		$constellations_{37}$	-1.51		
Files from *corpus I*					

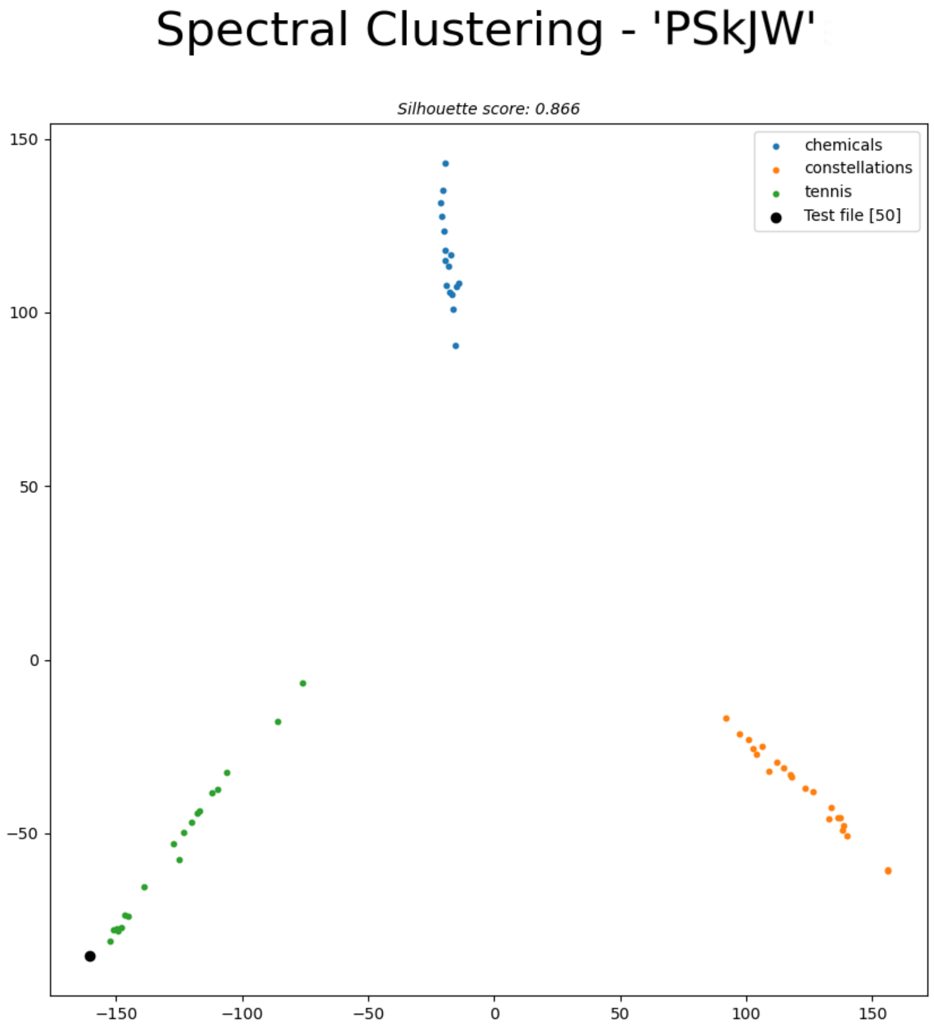


Figure G.1: Resulting clusters post test document classification during cross-validation – Test index 50 (Tennis *meta-class*). This plot was obtained with a matrix computed with $PSkJW$ metric and Spectral Clustering algorithm with 3 clusters (*corpus* I).

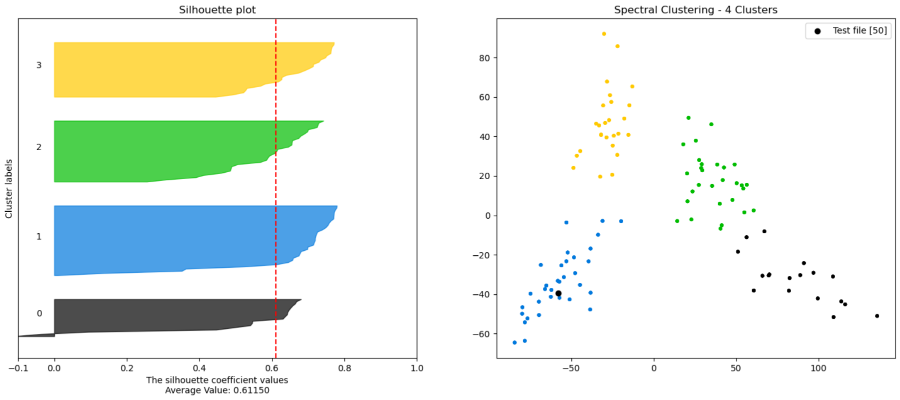


Figure G.2: Resulting clusters post test document classification during cross-validation – Test index 50 (Cats *meta-class*). This plot was obtained with a matrix computed with $PSkJW$ metric and Spectral Clustering algorithm with 4 clusters (*corpus* II).

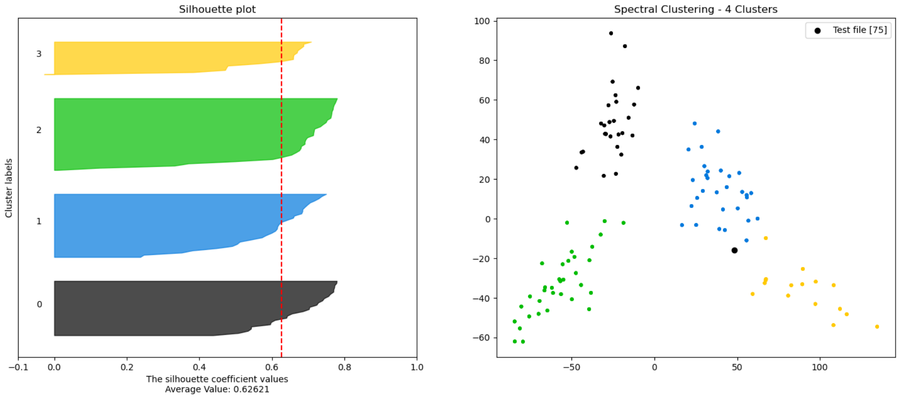


Figure G.3: Resulting clusters post test document classification during cross-validation – Test index 75 (Fish *meta-class*). This plot was obtained with a matrix computed with $PSkJW$ metric and Spectral Clustering algorithm with 4 clusters (*corpus* II). This document was one of the misclassified documents.

H

Cluster topic extraction tables

H.1 Expressions with size two or more (*corpus* I)

Extracted topics in order of importance from top to bottom (*corpus* I).		
Chemicals	Constellations	Tennis
chemical element with the symbol	astronomer Ptolemy	Grand Slam
atomic number	88 modern	Singles final was the championship
element with the symbol	remains one of the 88	Men's Singles
periodic table	48 constellations listed	Singles final was the championship tennis
Earth's crust	remains one of the 88 modern	final was the championship

Table H.1: Extracted expressions with Equation (3.13) where $G(RE) = 1$.

Extracted topics in order of importance from top to bottom (*corpus* I).		
Chemicals	Constellations	Tennis
chemical element with the symbol	astronomer Ptolemy	championship tennis match
atomic number	48 constellations listed	championship tennis
element with the symbol	88 modern constellations	Grand Slam
periodic table	2nd-century astronomer Ptolemy	Singles final was the championship tennis
alpha particles	International Astronomical Union	Singles final was the championship

Table H.2: Extracted expressions with Equation (3.13) where $G(RE) = Median(RE)$.

Extracted topics in order of importance from top to bottom (*corpus* I).		
Chemicals	Constellations	Tennis
alpha particles	48 constellations listed	championship tennis match
chemical element with the symbol	astronomer Ptolemy	Grand Slam
periodic table	88 modern constellations	Wimbledon Championships
atomic number	2nd-century astronomer Ptolemy	championship tennis
Manhattan Project	International Astronomical Union	storied Federer–Nadal

Table H.3: Extracted expressions with Equation (3.13) where $G(RE) = Sk(RE,3) \times Median(RE)$.

H.2 Expressions with size one (*corpus* I)

Extracted topics in order of importance from top to bottom (*corpus* I).		
Chemicals	Constellations	Tennis
beryllium	Orion	Wimbledon
actinium	represented	Championships
einsteinium-235	Centaurus	Australian
calcium	Triangulum	Djokovic
Einsteinium	astronomical	Federer–Nadal

Table H.4: Extracted expressions with Equation (3.15) where $G(t) = Length(t)$.

Extracted topics in order of importance from top to bottom (*corpus* I).		
Chemicals	Constellations	Tennis
actinium	Orion	Open
beryllium	star	Wimbledon
calcium	Draco	Nadal
boron	hare	Federer
bohrium	degrees	2014

Table H.5: Extracted expressions with Equation (3.15) where $G(t) = 1$.

H.3 Expressions with size two or more (*corpus* II)

Extracted topics in order of importance from top to bottom (*corpus* II).	
Birds	Cats
Union for Conservation of Nature	International Cat Association (TICA
Union for Conservation	Fanciers ' Association
Conservation of Nature	Cat Fanciers
eggs are laid	Cat Fanciers ' Association
Ancient Greek	Governing Council of the Cat Fancy
Fish	Dogs
Atlantic Ocean	sense of smell
Canary Islands	Bleu de Gascogne
aquarium trade	United States
dorsal fin	packs and descends
sexually dimorphic	Chien Français Blanc

Table H.6: Extracted expressions with Equation (3.13) where $G(RE) = 1$.

Extracted topics in order of importance from top to bottom (*corpus* II).

Birds	Cats
Union for Conservation	Fanciers ’ Association
Conservation of Nature	International Cat Association
monotonous mechanical insect-like reeling	Fédération Internationale Féline
internal parasites	Federation (WCF
worm Trichostrongylus tenuis	Governing Council
Fish	**Dogs**
Atlantic Ocean	Bleu de Gascogne
sexually dimorphic	sense of smell
Canary Islands	book The Intelligence
aquarium trade	Intelligence of Dogs
Union for Conservation	packs and descends

Table H.7: Extracted expressions with Equation (3.13) where $G(RE) = Median(RE)$.

Extracted topics in order of importance from top to bottom (*corpus* II).

Birds	Cats
Union for Conservation	Fanciers ’ Association
Conservation of Nature	Fédération Internationale Féline
worm Trichostrongylus tenuis	late-juvenile-onset neuromuscular degeneration
monotonous mechanical insect-like reeling	maintaining their kitten-like
internal parasites	playfulness into adulthood
Fish	**Dogs**
Union for Conservation	Bleu de Gascogne
Conservation of Nature	Chien Français Blanc
Canary Islands	book The Intelligence
Merlangius merlangus	Intelligence of Dogs
aquarium trade	sense of smell

Table H.8: Extracted expressions with Equation (3.13) where $G(RE) = Sk(RE, 3) \times Median(RE)$.

H.4 Expressions with size one (*corpus* II)

Extracted topics in order of importance from top to bottom (*corpus* II).

Birds	Cats	Fish	Dogs
warbler	Chartreux	Zosterisessor	Beagle-Harrier
Gymnocephalus	Highlander	ophiocephalus	Basset
bream	leopard	Micromesistius	Coonhound
bleak	Shorthair	goby	Anglo-Français
scoter	semi-long-haired	whiting	Vénerie

Table H.9: Extracted expressions with Equation (3.15) where $G(t) = Length(t)$.

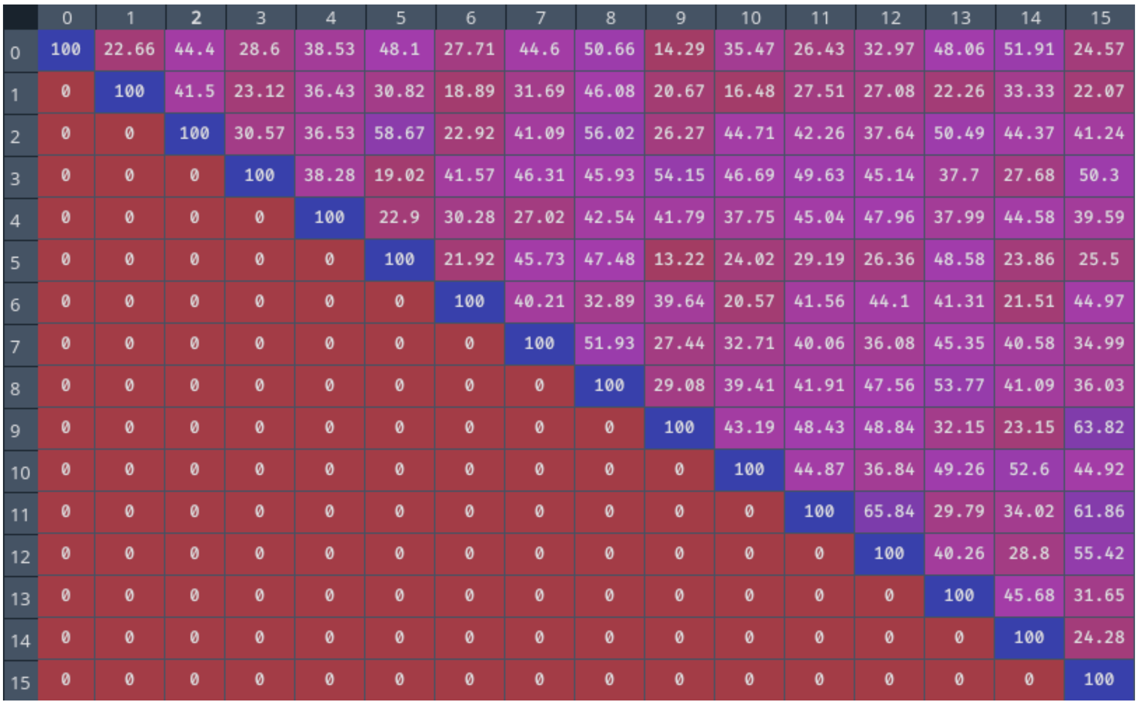
Extracted topics in order of importance from top to bottom (*corpus* II).

Birds	Cats	Fish	Dogs
tern	Manx	goby	Basset
warbler	leopard	whiting	Petite
bream	Chartreux	Zosterisessor	Vénerie
bleak	Highlander	ophiocephalus	Beagle-Harrier
scoter	Cymric	blenny	Coonhound

Table H.10: Extracted expressions with Equation (3.15) where $G(t) = 1$.

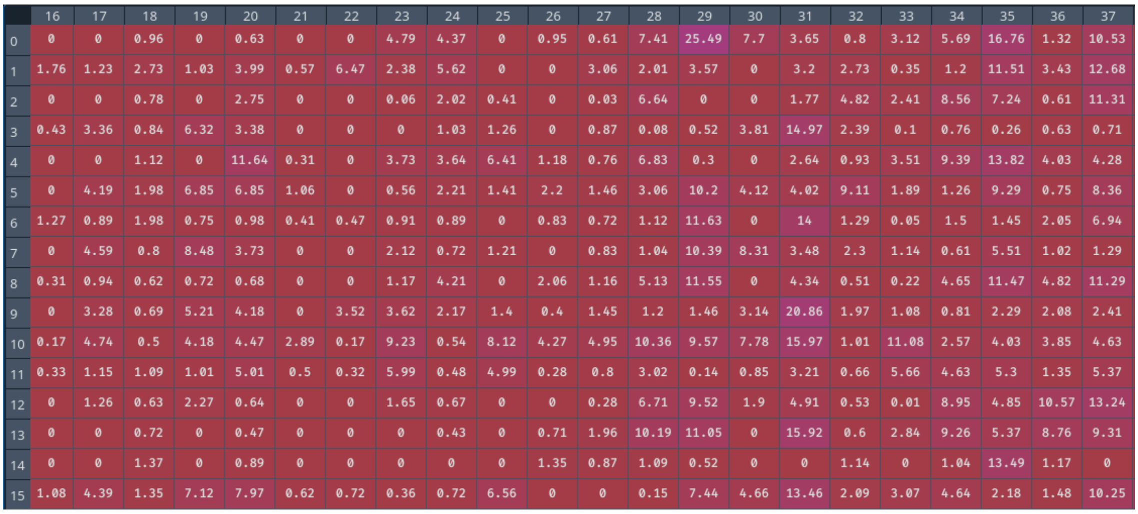
I

Results - Indirect Expressions



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	100	22.66	44.4	28.6	38.53	48.1	27.71	44.6	50.66	14.29	35.47	26.43	32.97	48.06	51.91	24.57
1	0	100	41.5	23.12	36.43	30.82	18.89	31.69	46.08	20.67	16.48	27.51	27.08	22.26	33.33	22.07
2	0	0	100	30.57	36.53	58.67	22.92	41.09	56.02	26.27	44.71	42.26	37.64	50.49	44.37	41.24
3	0	0	0	100	38.28	19.02	41.57	46.31	45.93	54.15	46.69	49.63	45.14	37.7	27.68	50.3
4	0	0	0	0	100	22.9	30.28	27.02	42.54	41.79	37.75	45.04	47.96	37.99	44.58	39.59
5	0	0	0	0	0	100	21.92	45.73	47.48	13.22	24.02	29.19	26.36	48.58	23.86	25.5
6	0	0	0	0	0	0	100	40.21	32.89	39.64	20.57	41.56	44.1	41.31	21.51	44.97
7	0	0	0	0	0	0	0	100	51.93	27.44	32.71	40.06	36.08	45.35	40.58	34.99
8	0	0	0	0	0	0	0	0	100	29.08	39.41	41.91	47.56	53.77	41.09	36.03
9	0	0	0	0	0	0	0	0	0	100	43.19	48.43	48.84	32.15	23.15	63.82
10	0	0	0	0	0	0	0	0	0	0	100	44.87	36.84	49.26	52.6	44.92
11	0	0	0	0	0	0	0	0	0	0	0	100	65.84	29.79	34.02	61.86
12	0	0	0	0	0	0	0	0	0	0	0	0	100	40.26	28.8	55.42
13	0	0	0	0	0	0	0	0	0	0	0	0	0	100	45.68	31.65
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	24.28
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Figure I.1: *PSkJW* matrix subsection of "Chemicals", computed with Indirect Expressions and 0.75 *second threshold* (*corpus* I).



	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
0	0	0	0.96	0	0.63	0	0	4.79	4.37	0	0.95	0.61	7.41	25.49	7.7	3.65	0.8	3.12	5.69	16.76	1.32	10.53
1	1.76	1.23	2.73	1.03	3.99	0.57	6.47	2.38	5.62	0	0	3.06	2.01	3.57	0	3.2	2.73	0.35	1.2	11.51	3.43	12.68
2	0	0	0.78	0	2.75	0	0	0.06	2.02	0.41	0	0.03	6.64	0	0	1.77	4.82	2.41	8.56	7.24	0.61	11.31
3	0.43	3.36	0.84	6.32	3.38	0	0	0	1.03	1.26	0	0.87	0.08	0.52	3.81	14.97	2.39	0.1	0.76	0.26	0.63	0.71
4	0	0	1.12	0	11.64	0.31	0	3.73	3.64	6.41	1.18	0.76	6.83	0.3	0	2.64	0.93	3.51	9.39	13.82	4.03	4.28
5	0	4.19	1.98	6.85	6.85	1.06	0	0.56	2.21	1.41	2.2	1.46	3.06	10.2	4.12	4.02	9.11	1.89	1.26	9.29	0.75	8.36
6	1.27	0.89	1.98	0.75	0.98	0.41	0.47	0.91	0.89	0	0.83	0.72	1.12	11.63	0	14	1.29	0.05	1.5	1.45	2.05	6.94
7	0	4.59	0.8	8.48	3.73	0	0	2.12	0.72	1.21	0	0.83	1.04	10.39	8.31	3.48	2.3	1.14	0.61	5.51	1.02	1.29
8	0.31	0.94	0.62	0.72	0.68	0	0	1.17	4.21	0	2.06	1.16	5.13	11.55	0	4.34	0.51	0.22	4.65	11.47	4.82	11.29
9	0	3.28	0.69	5.21	4.18	0	3.52	3.62	2.17	1.4	0.4	1.45	1.2	1.46	3.14	20.86	1.97	1.08	0.81	2.29	2.08	2.41
10	0.17	4.74	0.5	4.18	4.47	2.89	0.17	9.23	0.54	8.12	4.27	4.95	10.36	9.57	7.78	15.97	1.01	11.08	2.57	4.03	3.85	4.63
11	0.33	1.15	1.09	1.01	5.01	0.5	0.32	5.99	0.48	4.99	0.28	0.8	3.02	0.14	0.85	3.21	0.66	5.66	4.63	5.3	1.35	5.37
12	0	1.26	0.63	2.27	0.64	0	0	1.65	0.67	0	0	0.28	6.71	9.52	1.9	4.91	0.53	0.01	8.95	4.85	10.57	13.24
13	0	0	0.72	0	0.47	0	0	0	0.43	0	0.71	1.96	10.19	11.05	0	15.92	0.6	2.84	9.26	5.37	8.76	9.31
14	0	0	1.37	0	0.89	0	0	0	0	0	1.35	0.87	1.09	0.52	0	0	1.14	0	1.04	13.49	1.17	0
15	1.08	4.39	1.35	7.12	7.97	0.62	0.72	0.36	0.72	6.56	0	0	0.15	7.44	4.66	13.46	2.09	3.07	4.64	2.18	1.48	10.25

Figure I.2: *PSkJW* matrix subsection of similarities between documents of two different *meta-classes* (“Chemicals” and “Constellations”), computed with Indirect Expressions and 0.75 *second threshold*. The usage of Indirect Expressions has the drawback of similarities of documents of two distinct *meta-classes* rising (*corpus* I).

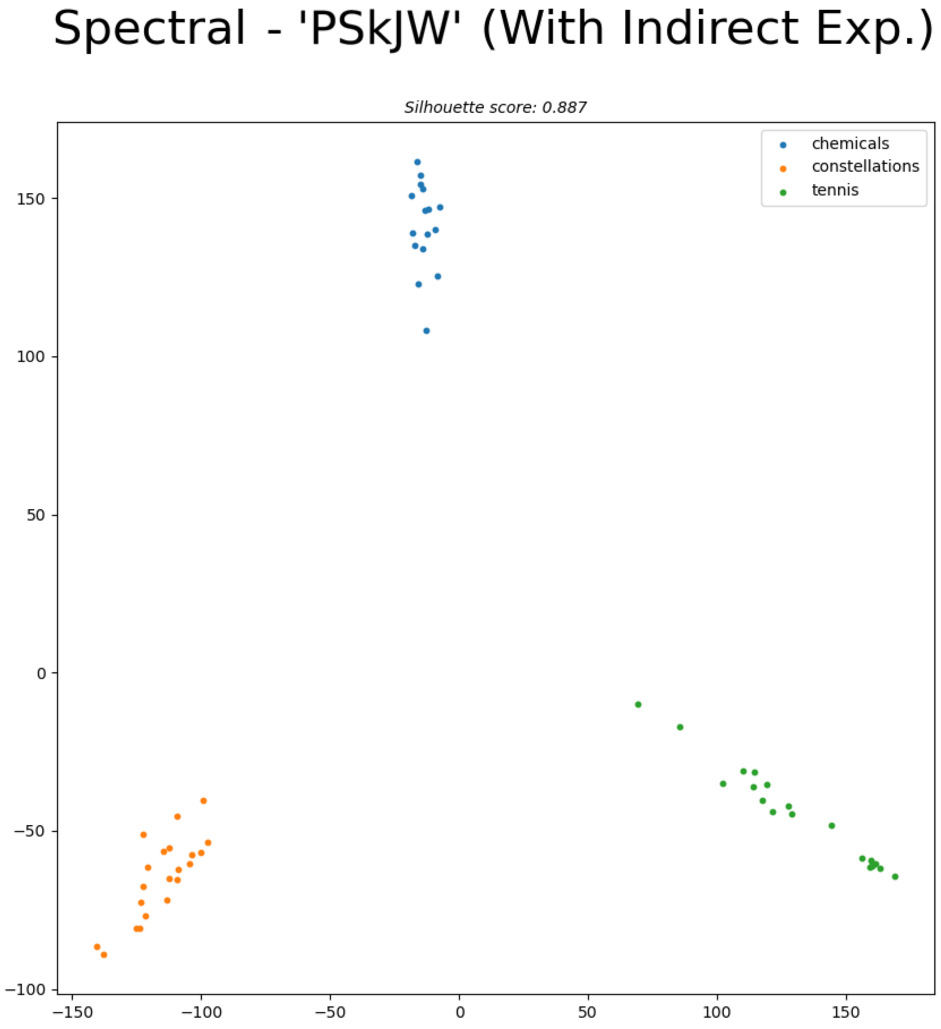


Figure I.3: Spectral Clustering results with *PSkJW* metric, with Indirect Expressions and 0.75 *second threshold* (*corpus* I).

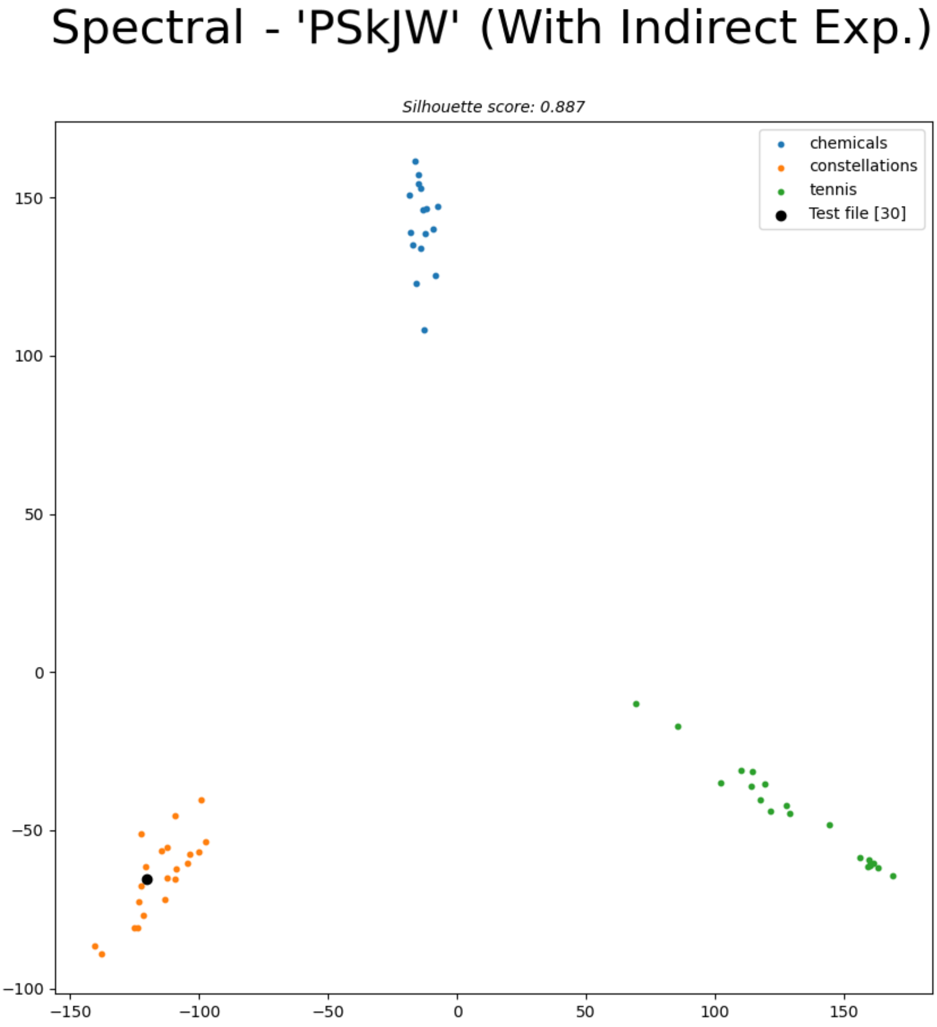


Figure I.4: Spectral Clustering results with $PSkJW$ metric, with Indirect Expressions and 0.75 *second threshold*. The black marker represents the transformed coordinates of test document with index 30 in *corpus I*.

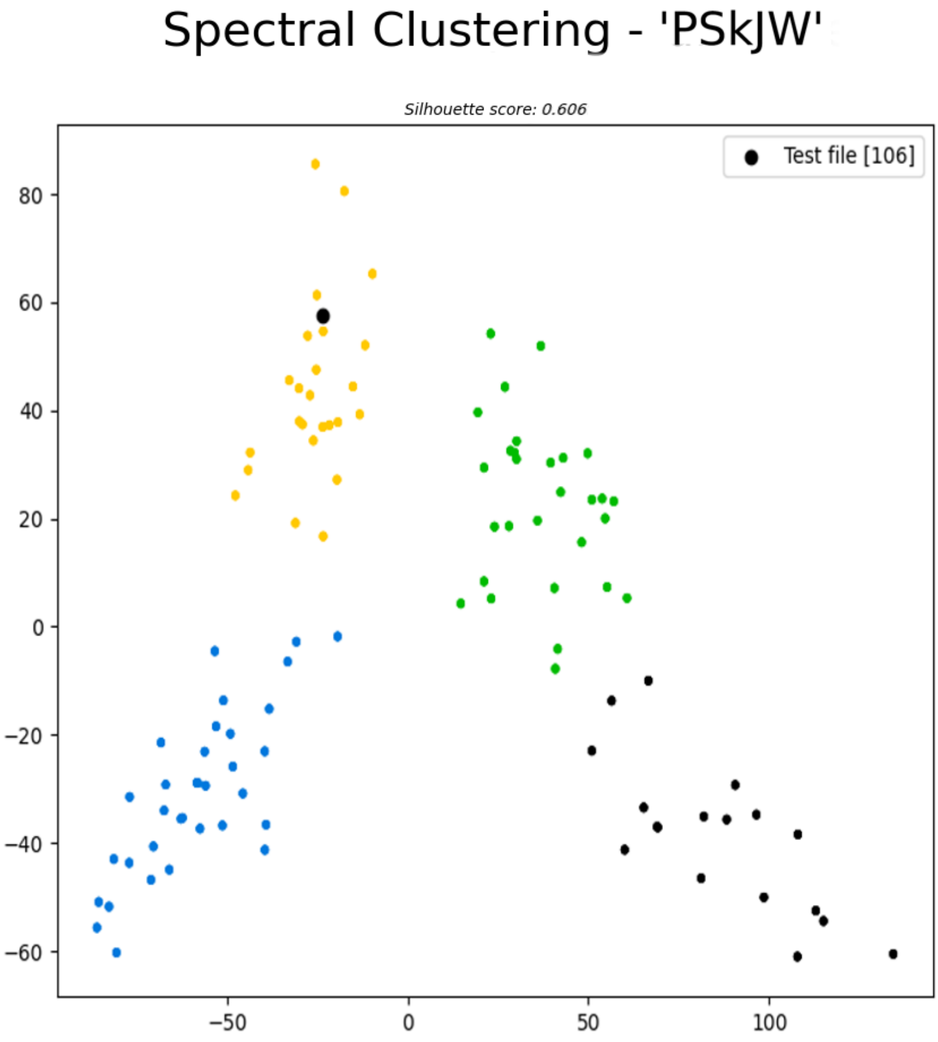


Figure I.5: Spectral Clustering results of *corpus* II with $PSkJW$ metric only. The black marker denotes the coordinates of the document of index 106.

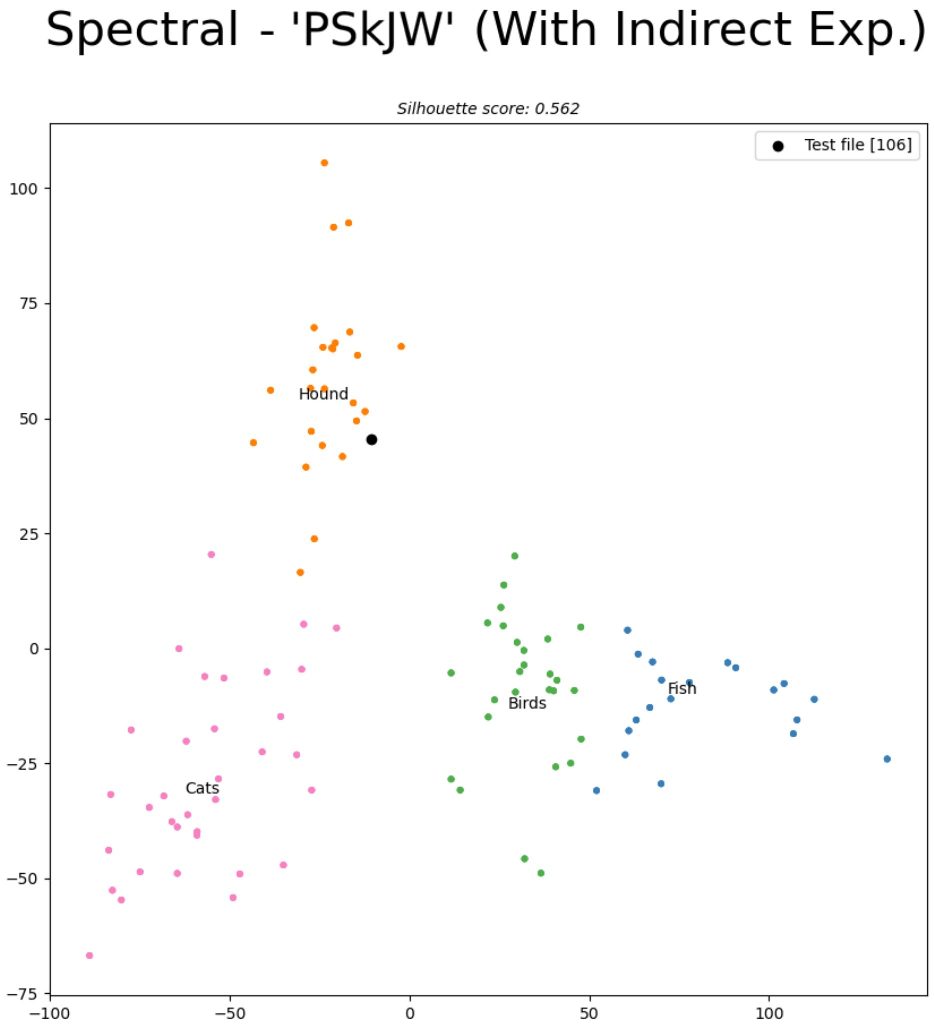


Figure I.6: Spectral Clustering results with *PSkJW* metric, with Indirect Expressions computed with 0.89 *second threshold*, applied to *corpus* II – The black marker denotes the coordinates of test document of index 106 (worth noting that the “Hound” *meta-class* is the same as “Dogs”).

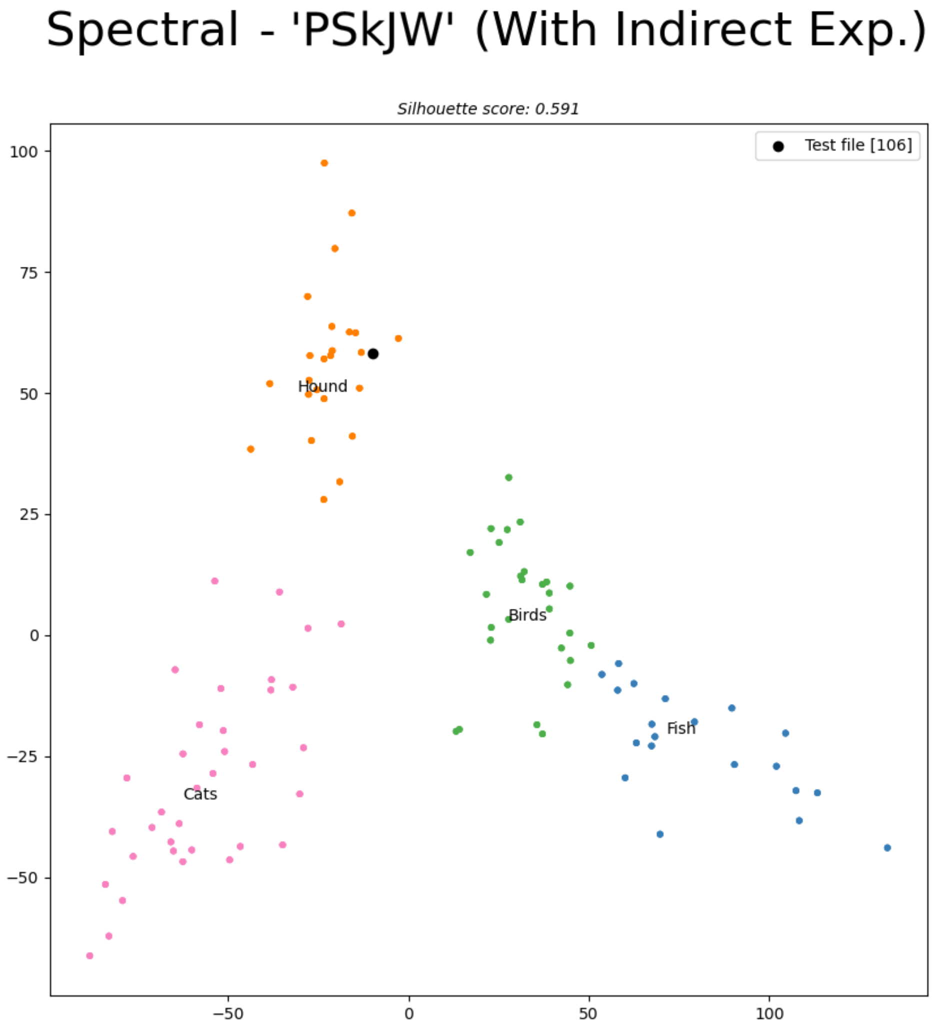


Figure I.7: Spectral Clustering results with *PSkJW* metric, with Indirect Expressions computed with 0.9 *second threshold*, applied to *corpus* II – The black marker denotes the coordinates of test document of index 106 (Worth noting that the “Hound” *meta-class* is the same as “Dogs”).

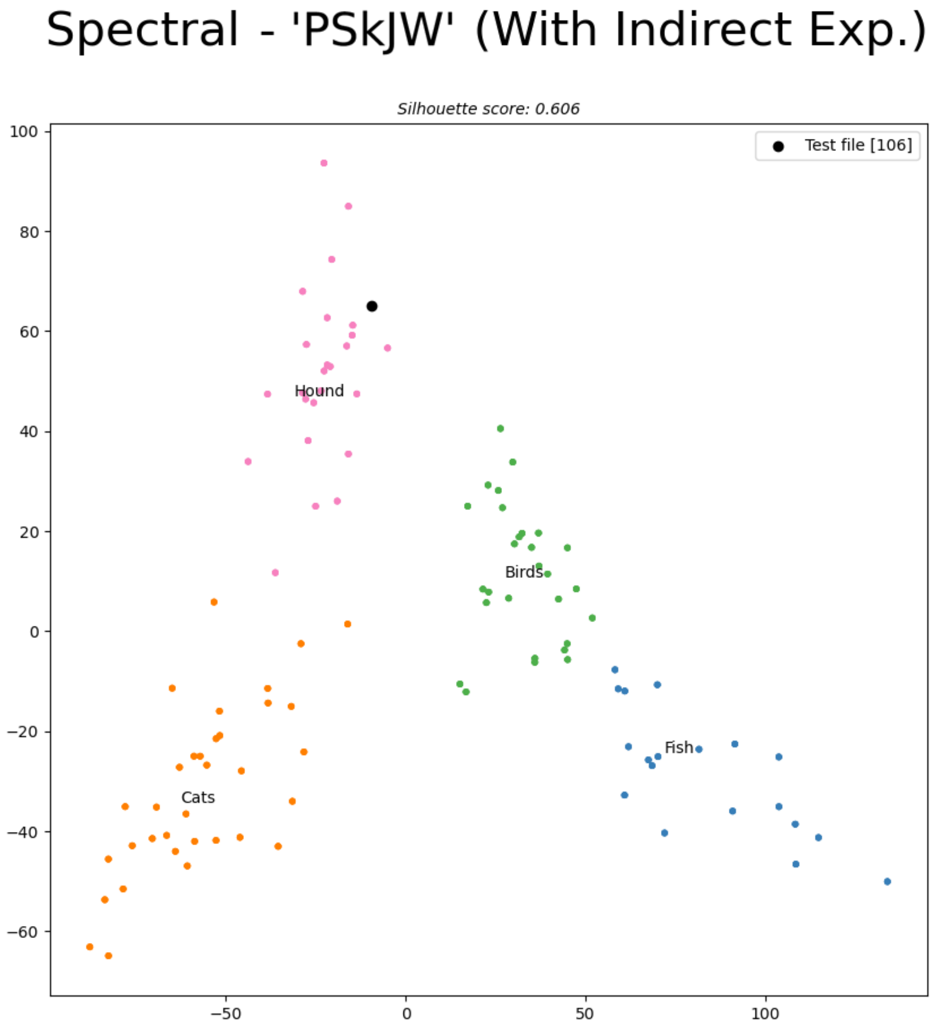


Figure I.8: Spectral Clustering results with *PSkJW* metric, with Indirect Expressions computed with 0.91 *second threshold*, applied to *corpus* II – The black marker denotes the coordinates of test document of index 106 (Worth noting that the "Hound" *meta-class* is the same as "Dogs").

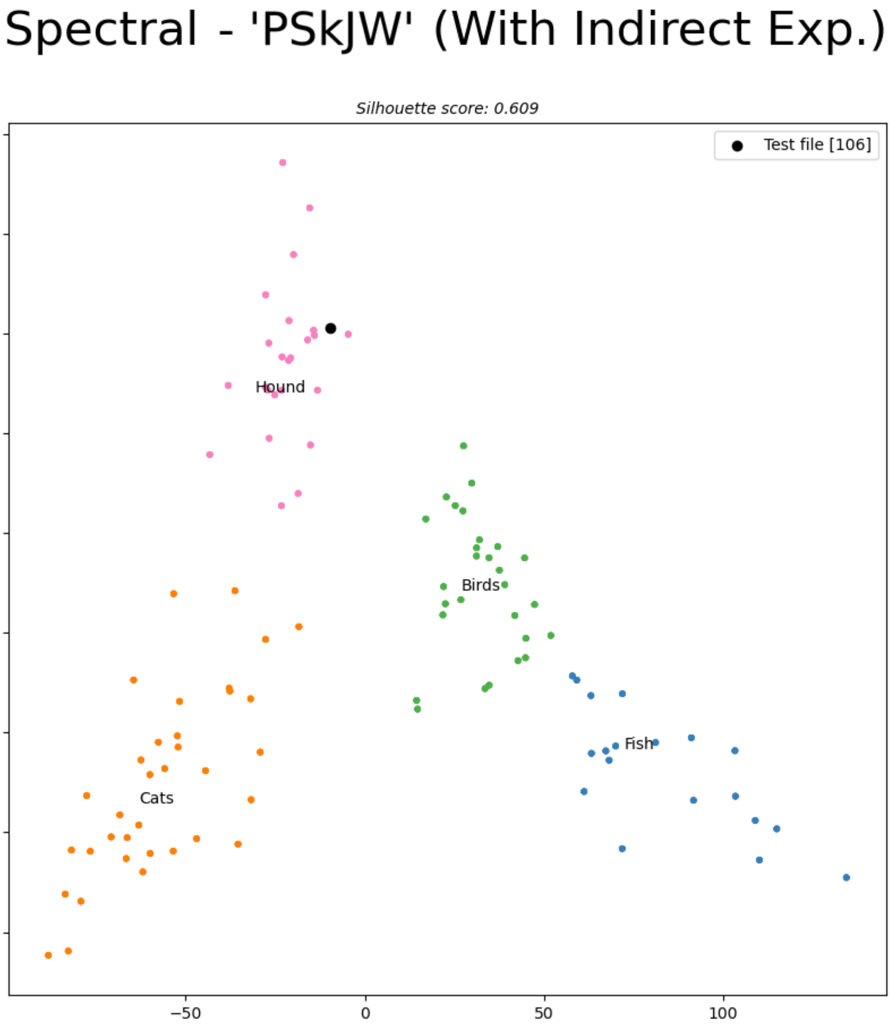


Figure I.9: Spectral Clustering results with *PSkJW* metric, with Indirect Expressions computed with 0.9045 *second threshold*, applied to *corpus* II – The black marker denotes the coordinates of test document of index 106 (Worth noting that the “Hound” *meta-class* is the same as “Dogs”). This was the *second threshold* that maximized the obtained Silhouette Score.