Document Simplicial Complex

Varun Mishra

A Thesis Submitted to

Indian Institute of Technology Hyderabad

In Partial Fulfillment of the Requirements for

The Degree of Master of Technology

Department of Computer Science and Engineering

June 2018

Declaration

I declare that this written submission represents my ideas in my own words, and where ideas or words of others have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and can also evoke penal action from the sources that have thus not been properly cited, or from whom proper permission has not been taken when needed.

(Signature)

Varun Mishra

(Varun Mishra)

CS16MTECH11018

(Roll No.)

Approval Sheet

This Thesis entitled Document Simplicial Complex by Varun Mishra is approved for the degree of Master of Technology from IIT Hyderabad

(Dr. Manohar Kaul) Adviser
Dept. of Computer Science
IITH

Dr. Srijith (Examiner)

Dr. Maunendra (Examiner

Dr. Vineeth (chairman)

Acknowledgements

Firstly, I would like to express my sincere gratitude to my advisor Dr. Manohar Kaul for the continuous support of my master's study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis.

I would also like to thank all the faculy members in CSE and Mahemaics department for their support and encouraging behaviour and pushing me to do always better.

Dedication

Dedicated to teachers, family and friends.

Abstract

A **k-simplex** is defined as k-dimensional geometric structure which is the convex hull of $k+1$ points. Given $k+1$ points $x_0, ..., x_k \in \mathbb{R}^k$ which are affinely independent, the set

$$\mathcal{C} = \left\{ a_0x_0 + ... + a_kx_k \middle| \sum_{i=0}^{k} a_i = 1 \text{ and } a_i \geq 0 \text{ for all } i \right\},$$

is defined as the k-simplex determined by them. Simplex is a very basic building structure in abstract topology. Collection of simplexes (or simplices) under certain condition is called geometrical simplicial complex, which further helps to analyze a geometrical structure on bigger scale. An abstract simplicial complex is a purely combinatorial description of the geometric notion of a simplicial complex, consisting of a family of non-empty finite sets closed under the operation of taking non-empty subsets.

A text document can be visualized as a geometric structure in topology. A document is defined as a collection of words, where each word is considered to be a part of vocabulary having a certain meaning. And an n-gram is a contiguous sequence of n items from a given sample of text. Using the n-gram concept to define a simplex we can construct an abstract simplicial complex out of every text document. Thus from this model, every simplex catches the local structure or behavior while a document simplicial complex, which is the collection of all n-1 simplex, captures the global behavior of the document. We will study this considering we have a bag of documents i.e. the universal set of documents.

The aim of this thesis is to understand abstract structure admitted by text documents to find more accurately the similar documents from the given family if text documents. In our discussion, we will visualize a document as a geometrical entity and will make use of such representation of a text document to fast the process of querying, where given a query document one can find the semantically similar documents more efficiently in the sense of time and similarity. For example, given a set of documents as {1. "after clearing high school one joins college", 2. "College can be joined only after passing high school" and 3. "High school and college must be attended by everyone"} the document 1 and 2 are more semantically similar that 1 and 3 or 2 and 3.

After a brief glance at abstract topology, we study the topological structure and behavior of text documents. A novel representation of documents is given in this thesis. Using this new structure of a text document we represent each document as a geometrical entity which further can be analyzed using topological tools. Using Earth

Mover's distance and Hausdorff distance we give a new formulation to fetch semantic documents for a given query. To represent documents as a mathematical structure in some $\mathbb{R}^k$, we use Word2Vec model to find vector representation of each word in a text document.

Contents

Chapter 1

Introduction

General topology tries to abstract out the notion of “distance” on $\mathbb{R}$. Indeed, a large part of point-set topology studies metric spaces. it turns out to define continuity, we don’t need to know much about real numbers. This enables us, for example, to do analysis with graphs, which are thoroughly discrete objects. Abstract topological spaces take this to a greater height you actually don’t need a metric either. Just defining which elements of X are “close” is enough, you don’t need to know how close they are. That said, you pick a subset τ of $P(X)$ satisfying some desirable properties, where elements of τ are the subsets of X consisting of elements “close” to each other. The members of τ are said to be the open subsets of X.

Topology has many applications since it is concerned with the properties of space that are preserved under continuous deformations. From given pool of documents, and a query our task is to find the most relevant document which is semantic similar to query document. To do so we will use tools from abstract topology in a manner to find the nearest document with same structure and representation. Simplex which is the convex hull of affinely independent points in given Euclidean space would be used to capture and represent the local structure of document i.e. on a word and phrase level. While the simplicial complex builds on the top of simplices will capture the global behavior of document, which means the flow in document i.e. how the whole text is organized in a document.

1.1 Basic Idea

Topological view for a text document has never been explored, while it can give us various insights for texts, it also gives us a new representation for text. Using word2vec we can project each vocabulary term to higher dimensional Euclidean space

$\mathbb{R}^k$, where each word is clustered with semantic similar words. Now what we want is the same result for documents i.e. such cluster for text documents where documents are clustered by their semantic meaning. So, to achieve that we use various tools and terminologies of abstract topology, We give a novel structure associated with every document. We also propose a novel metric to define the distance between two such representations, which measure, in turn, the semantic difference between documents. Here we try to achieve all these results while maintaining the efficiency. Nearest neighborhood (kNN) is used to evaluate our proposed metric and representation, where we compete with state of the art methods such as word mover's distance, LSI, LDA, mSDA etc.

1.2 Chapter wise organization

Chapter 2 In this chapter we will survey related work. Mathematical model of some methods is also defined, we will focus on WMD which is currently state of the art method.

Chapter 3. This part of the thesis contains basic building blocks of our proposed method. Description of different tools used from topology will be provided in this chapter with ample example to make it easy to understand for other readers.

Chapter 4. How are we relating a text document to a topological structure is discussed in this chapter. Further, we will discuss the behavior of such representation and show some example on small documents.

Chapter 5. This chapter will discuss the main feature to establish a semantic difference between two document topological structure i.e. distance. We will look into Earth Mover's distance and Hausdorff distance as these two will be used to build our main distance model to find the semantic difference between two document structure.

Chapter 6. We will compare the result of our proposed method with state of the art method on classification task. Using k-NN method we will achieve such comparison, and will graphically show the results.

Chapter 7. In this final chapter, we will discuss some possible points for our future work and conclude the present work.

Chapter 2

Related Work

This chapter surveys previous work done in document distance and the methods developed by researchers that we will be using to compare our metric. As the distance between document is closely tied with learning new representation, we will be discussing such methods also.

Comparison among methods will be done by kNN classification error, to do that, we will use different metrices for different representations. In following sections we will learn various methods to represent a text document. Generative models, count based models are some popular models to represent a text.

Dataset plays a vital role to test a method. We evaluate our approach on supervised document datasets. Stats of each dataset and train test split and other details will be discussed in this chapter.

2.1 Introduction

In this section, we will describe various methods and representation approach for a text document. Bag-of-word model in which a text is represented as the multiset of its words, irrespective of grammar and word order, but only retaining frequency. One of the first works to systematically approach the document representation is done using term frequency and inverse document frequency (tf-idf) and still quite popular, one of the famous variants is Okapi BM25 function[1], which describe a score for each (word, document) pair and is designed for ranking applications.

Latent Dirichlet Allocation (LDA)[2] is a generative statistical model, that assumes a Dirichlet prior over the latent topics. While latent semantic indexing (LSI)[3] learns latent topics by performing a matrix decomposition (SVD) on the term-document matrix.

Methods that learn document representation using deep learning include Stacked Denoising Autoencoders (SDA)[4] and newer and faster version marginalized SDA (mSDA)[5], this method learns the work correlation via dropout noise in stacked neural networks.

Recently proposed metric, word mover's distance depends on word embeddings, to get the best results author used Google News word2vec model[1]. This method uses word embedding and represent each document as a collection of these embedded words and using earth mover's distance (EMD)[2], finds the similar documents.

2.1.1 Bag-of-Words Model

The BOW model is a simple representation used in natural language processing and information retrieval. This model is also known as vector space model, in this, the document is represented as a vector of words present in it. This model disregard the grammar and order of words in the document which ignores the context and in turn meaning of words. Thus not a very good model for the classification task.

2.1.2 Term Frequency Model

Word frequency count in document used in creating a vector instead of only binary scoring. But, this makes model bias towards words which are frequent and so rescale the frequency inverse document frequency is used. In this model, term frequency is a scoring of the word in document and inverse document frequenct is a scoring of how rare the word is across documents.

$$idf(t, D) = \log \frac{N}{1 + |\{d \in D : t \in d\}|}$$

$$tf\text{-}idf(t, d) = tf(t, d) * idf(t, D)$$

$|\{d \in D : t \in d\}|$ is the number of document where term t appears. N is total number of document in corpus D. t is a term in document d.

Okapi BM25[1] (BM stands for best matching) is ranking function used to rank the documents according to given query. This method extends the tf-idf for each term t in a document d. Given a query Q containing words $q_1, \ldots, q_n$, the BM25 score of

[1] Explained in chapter 3

[2] Explained in chapter 5

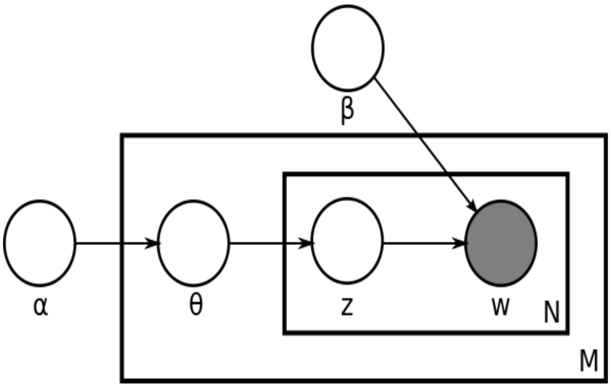

Figure 2.1: plate notation representing the LDA model [6]

a document d is:

$$score(d, Q) = \sum_{i=1}^{n} BM25(q_i, d)$$

$$BM25(t, d) = \frac{idf(t, D)tf(t, d)(k_1 + 1)}{tf(t, d) + k_1(1 - b + b\frac{|D|}{D_{avg}})}$$

D_{avg} is the average document length in the corpus. k_1 and b are free paramenters.

2.1.3 Latent Semantic Indexing

LSI[3] assumes that words that are close in meaning will occur in documents which are similar. To describe the occurrences of terms in documents we use the term-document matrix which is a sparse matrix where rows and columns correspond to terms and documents respectively. To make the high rank, sparse term-document matrix, a low-rank matrix SVD is used which also preserves similarity structure among columns. This yields a new representation for each document in the collection. Queries will also be represented in the low-rank manner which makes us able to compute query-document similarity score in this such representation. This process is known as latent semantic indexing.

2.1.4 Latent Dirichlet Allocation

This is a generative statistical model that represents a document as a distribution over word topics. Each document can be viewed as a mixture of various topics, where the topic distribution is considered to be a Dirichlet prior. Figure 2.1 shows the plate notation which captures the dependecies among the many variables. We consider M

documents each of size N_i and vocabulory size V, then the generative process for each word w_j in document d_i is as follows:

1. Choose $\theta_i \sim Dir(\alpha)$, where $i \in \{1 \ldots, M\}$ and α is a symmetric parameter of Dirichlet prior on per document topic distributions, θ_i is the topic distribution for document i

2. Choose $\phi_k \sim Dir(\beta)$, where $k \in \{1, \ldots, K\}$ and β is the parameter of the Dirichlet prior on the per topic word distribution, ϕ_k is the word distribution for topic k,

3. For each of the word positions i, j, where $i \in \{1, \ldots, M\}$, and $j \in \{1, \ldots, N_i\}$

   (a) Choose a topic $z_{i,j} \sim Multinomial(\theta_i)$. $z_{i,j}$ is the topic for the j-th word in document i,

   (b) Choose a word $w_{i,j} \sim Multinomial(\phi_{z_{i,j}})$, $w_{i,j}$ is the specific word.

2.1.5 Marginalised Stacked Denoising Autoencoder

A deep learning method, where representation for each document is learned from stacked denoising autoencoder (SDAs) and marginalized for fast training[5]. For sentiment analysis in documents, the SDAs have shown the state of the art performance[4].

2.1.6 Word Mover's Distance

WMD[7] is a distance function between text documents which is based on word embeddings that learns semantically meaningful representation for words. This distance function measures the dissimilarity between two documents by using earth mover's distance that is by calculating the minimum amount of work done to convert one embedded document to other.

Given word2vec embedding matrix $X \in \mathbb{R}^{d \times n}$, where each column $x_i \in \mathbb{R}^d$ represents the embedding of the i^{th} word in d-dimensional space for vocabulary size n. The distance between word i and word j is given by $c(i, j) = ||x_i - x_j||_2$. Moreover, if word i appears c_i times the normalised weight is give by $d_i = \frac{c_i}{\sum_{j=1}^{n} c_j}$. If **d** and **d'** are two documents and let $\mathbf{T} \in \mathbb{R}^{n \times n}$ be a sparse flow matrix where $\mathbf{T}_{ij} \geq 0$ denotes how much of word i in **d** travels to word j in **d'**. Then, the WMD between two documetss is defined as minimum cumulative cost required to move all words from **d** to **d'**, while ensuring the entire ongoing and outgoing flow to match the total weight of each word.

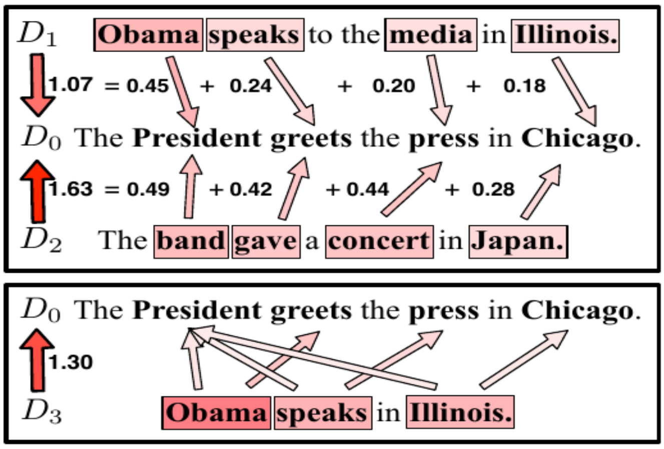
D_1 Obama speaks to the media in Illinois.
1.07 = 0.45 + 0.24 + 0.20 + 0.18
D_0 The President greets the press in Chicago.
1.63 = 0.49 + 0.42 + 0.44 + 0.28
D_2 The band gave a concert in Japan.
D_0 The President greets the press in Chicago.
1.30
D_3 Obama speaks in Illinois.

Figure 2.2: (Top:)The components of the WMD metric between a query D_0 and two sentences D_1, D_2 (with equal BOW distance). The arrows represent flow between two words and are labeled with their distance contribution. (Bottom:) The flow between two sentences D_3 and D_0 with different numbers of words. This mismatch causes the WMD to move words to multiple similar words [7]

Foramlly, it can be given as solution for following optimization problem, which is a special case of earth mover's distance and it is also a well studied problem. To highlight this connection authors refer the resulting metric as *word mover's distance.*

$$\min_{T\geq 0} \sum_{i,j=1}^{n} \mathbf{T}_{ij} c(i,j)$$

$$\text{with constraints :} \sum_{j=1}^{n} \mathbf{T}_{ij} = d_i \quad \forall i \in \{1,\ldots,n\}$$

$$\sum_{i=1}^{n} \mathbf{T}_{ij} = d'_j \quad \forall j \in \{1,\ldots,n\}.$$

Chapter 3

Simplicial Complex

Visualization of a document in a topological structure is a new representation, which enables us to get insight of a document in several levels i.e. local, global, topic level etc. Before moving to the study of document simplicial complex, in this chapter we study the building blocks for this concept and will also cover some other fundamental topics that we need to accomplish our goal.

To establish a relationship between a text document to a geometrical entity, where each word or phrase will be a node, we must have a relation between word and geometric coordinates in Euclidean space. For that, we used *word2vec* model, as the name suggests each word in this model is associated with a vector. Also, this model maps the word to a semantically meaningful space, where similar words are near to each other. The vectors are very good at answering analogy questions of the form "a is to b as c is to ?". For example, man is to woman as uncle is to? (aunt).

3.1 Simplicial Complex

In this section, basic concepts and terminology from abstract topology are explained which we will need for further discussion. Many of the concepts and visualizations are taken from [8]

Definition 1. *Given a set $\{a_0, \dots, a_n\}$ of points in $\mathbb{R}^N$, this set is said to be geometrically independent if for any real scalars t_i, the equations*

$$\sum_{i=0}^{n} t_i = 0 \quad \textit{and} \quad \sum_{i=0}^{n} t_i a_i = 0$$

imply that $t_0 = t_1 = \cdots = t_n = 0$.

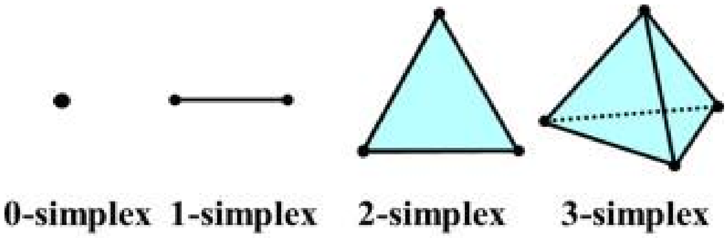

Figure 3.1: Graphical illustration of n-simplices

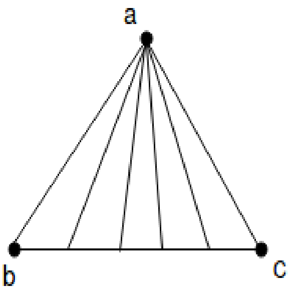

Figure 3.2:

It can be observed that in genearl $\{a_0, \ldots, a_n\}$ is geometrically independent if and only if the vectors $\{a_1 - a_0, \ldots, a_n - a_0\}$ are linealy independent.

So, it is clear that one-point set is always geometrically independent. Two distinct points in $\mathbb{R}^N$ form a geometrically independent set, as do three non-collinear points, four non-coplanar points.

Definition 2 (Simplex)**.** *Let $\{a_0, a_1, \ldots, a_n\}$ be a geometrically independent set in $\mathbb{R}^N$. The n-simplex σ spanned by $a_0, \ldots, a_n$ is the set of all points x in $\mathbb{R}^N$ such that,*

$$x = \sum_{i=0}^{n} t_i a_i.$$

Where $\sum t_i = 1,\ t_i \geq 0,$ for all i. The scalars t_i are uniquely determined by x; they are called the barycentric coordinates of the point x of σ with respect to $a_0, \ldots, a_n$.

Figure 3.1 shows graphically how the n-simplices look in 3-dimensional space.

Note 1. *It can be easily observed that the k-simplex spanned is the union of all line segments joining a_0 to points of the simplex spanned by $a_1, ..., a_k$. Two such line segment intersect only at a_0. Refer Figure 3.2*

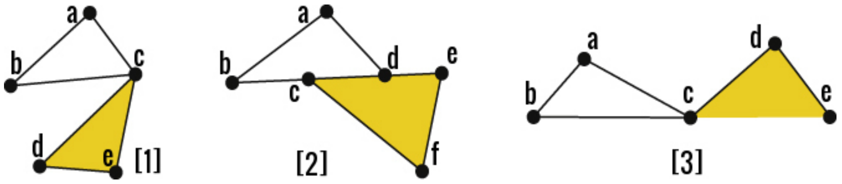

Figure 3.3: [2] and [3] are not the simplicial complexes while [1] is.

Definition 3 (face)**.** *Given the set of points $\{a_0, \ldots, a_n\}$ in $\mathbb{R}^N$, let the spanned simplex be σ. Any simplex spanned by a subset of $\{a_0, \ldots, a_n\}$ is called a face of σ. The faces of σ different from σ itself are called the proper faces.*

1. any singleton subset of σ, is a 0-face of σ.

2. set $\{k_i, k_{i+1}\}$ spans a 1-face.

3. set $\{k_i, \ldots, k_{i+m}\}$ spans a m-face of simplex.

Definition 4 (Boundary and Interior)**.** *The boundary of a simplex σ is defined as the union of all its proper faces. Boundary of σ is denoted as $\partial\sigma$. Interior of simplex is denoted as Int σ and defined as Int $\sigma = \sigma - \partial\sigma$.*

Definition 5 (Simplicial Complex)**.** *A simplicial complex K in $\mathbb{R}^N$ is defined as the collection of simplices in $\mathbb{R}^N$ such that following properties hold.*

1. *Every face of a simplex of K is also in K.*

2. *The nonempty intersection of any two simplices of K is a face of each of them.*

Example: Consider the figure 3.3[1], which is the collection of simplices spanned by $\{a, b\}$, $\{b, c\}$, $\{c, a\}$ and $\{c, d, e\}$. As each face of these simplicies is also an element of collection and the intersection are just the vertices, which are face for both of the simplicies. So, the collection is a simplicial complex. On the other hand figure 3.3[2] is not a simplicial complex. It is the collection of simplices spanned by the set $\{a, b\}, \{b, d\}, \{d, a\}$ and $\{c, e, f\}$, and this collection does not satisfy the second condition for being a simplicial complex as the intersection of two simplices spanned by $\{b, d\}$ and $\{c, e, f\}$ is the line segment spanned by $\{c, d\}$ which if not a face for any of the intersecting simplices. As for figure 3.3[3], it is again not a simplicial complex, as it does not follow the first rule. It is a collection of simplicies spanned by $\{a, b\}$,$\{b, c\}$, $\{c, a\}$ and $\{c, d, e\}$, but the face spanned by $\{c, e\}$ is not included in the collection. So 3.3[3] is not a simplicial complex.

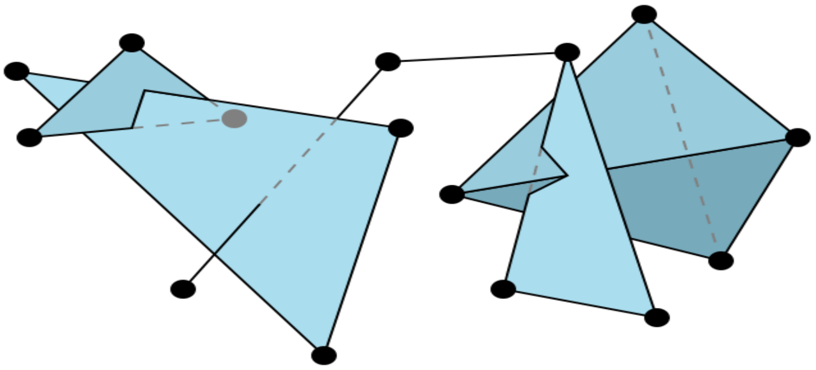

Figure 3.4: An ASC which is not a simplicial complex [9]

Definition 6 (p-skeleton)**.** *A subcollection L, of simplicial complex K such that it is a simplicial complex on its own, is called subcomplex of K. Subcomplex, such that it is the collection of all simplices of K of dimension at most p; is called the p-skeleton of K and is denoted by $K^{(p)}$. The points of the collection $K^{(0)}$ are called the vertices of K.*

Definition 7 (Abstract Simplicial Complex)**.** *An abstract simplicial complex (ASC) is a collection C of finite non-empty sets, such that if A is an element of C, so is every nonempty subset of A.*

Remark: The element A of C is called a simplex of C, the dimension is one less than the number of its element. Each nonempty subset of A is called a face of A.

Note: Every geometrical simplicial complex is an ASC but not every ASC is geometric simplicial complex.

3.2 Word2Vec Model

Word2Vec [10] is a celebrated word embedding model that learns representation for words and thus maps a word to a high dimensional Euclidean space. To make a text document understandable for machines we need them to be a mathematical entity, and this is we get for our method by using word2vec model.

Tom Mikolov et.al. [10, 11], introduces the continuous Bag-of-Words (CBOW) and Skip-Gram model architecture to produce a representation of words. Tf-idf representations and scores give us an idea of word's importance in a document but not really capture the semantic meaning. Word2Vec is a neural network model that given an unlabelled training corpus, projects each word to a unique vector that encodes its semantic meaning. The vectors are very good at answering analogy questions of the form “a is to b as c is to ?”. For example, consider the analogy “Woman is to queen as man is to king”. It turns out that

$$v_{queen} - v_{woman} + v_{man} \approx v_{king}$$

where $v_{queen}, v_{woman}, v_{man}$ and v_{king} are word vectors for *queen, woman, man* and *king* respectively.

The model tries to maximize the conditional log probability for a word to be the neighbouring word of the input word i.e. given a word sequence $w_1, \ldots, w_n$,

$$\frac{1}{n}\sum_{i=1}^{n}\sum_{j \in nbd(i)} \log p(w_j|w_i)$$

where $nbd(i)$ is the set of neighbouring words of word w_i and $p(w_j|w_i)$ is the hierarchical softmax of the associated word vectors v_{w_j} and v_{w_i}.

Chapter 4

Document Simplicial Complex

In the last chapter, we covered some background work to define our model. For a text, how we define a simplex and top of that what do we mean by a document simplicial complex, these questions will be answered in this chapter. Association of the text to a mathematical entity, this is what we need to get our model. Such association at the different level will be covered in this chapter. From a word to a phrase to a whole document can be transformed to a geometrical structure to find beautiful insights from a text.

Simplicial complex and abstract simplicial complex are the tools we need to focus on while defining document simplicial complex. Simplicial complex, which gives the geometric insight and visualization of points, will give us high dimensional geometrical structure of a document while abstract simplicial complex associated with a text document gives an embedding of a document in high dimensional space with no such visual insights and it will be used to get a mathematical aspect of a text.

First, to get an association of each word with some vector of high dimensional space we use word2vec model. Word embedding with semantic meaning is needed to define a structure for text that will be used to fetch similar documents. Further, n-gram model used to define a relation among text and simplices, n-gram model makes us able to capture the order of words and phrases in the document.

In chapter 2, we covered the related work and also described some of the methods that we will be comparing our model with. Word Mover's distance (WMD), which is a variant of earth mover's distance (EMD) for a document, is state of the art method, but in this model, the structure of a document loses the order of words. In our method, to capture the ordering of words we used n-gram model.

4.1 n-gram Model

Given a sequence of n words $w_0, \ldots, w_n$ what is the $n+1$th word, this is what n-gram model is, predicting the next word given the history and a sequence of words. This model formalizes the intuition of next word by introducing models that assign a probability to each possible next word. And, that can also be used to assign a probability to an entire sentence.

An n-gram is a contiguous sequence of n *words from a given sample of text.* So, given a sentence as *I noticed two BMW's on road.*, the contiguous sequence of 2 words: "I noticed", "noticed two", "two BMW's" and so on are the 2-grams or formally said the *bigrams.* Similarly the sequences "I noticed two", "noticd two BMW's" and so on are the 3-grams or *trigrams.*

With a large enough corpus, such as the web, we can estimate the probability of a word w, given the history h, i.e. $p(w|h)$. One way to estimate such probability is,

$$p(w|h) = \frac{p(w,h)}{p(h)},$$

while it seems a rather easy estimation but it turns out to be quite irksome to do so, as to calculate the joint probability for a whole sentence and a word asks a lot to estimate.

So to estimate the probability of a given word and history we can rely on conditional probability of *bigrams* rather than for joint probability of a whole sentence and a word. To put it formally, let the given sequence of words be $h = (w_0, \ldots, w_{n-1})$ and lets denote the sequence of n words as w_1^n then we want to estimate the $p(w_n|h)$. In other words we compute $p(w_0, \ldots, w_{n-1}, w_n)$. Using the chain rule of probability:

$$p(w_1^n) = p(w_1)p(w_2|w_1)p(w_3|w_1^2)\ldots p(w_n|w_1^{n-1})$$

$$= \prod_{k=1}^{n} p(w_k|w_1^{k-1})$$

The bigram model approximate the probability of a word given all the previous words $p(w_n|w_1^{n-1})$ by using only the conditional probability of the preceding word $p(w_n|w_{n-1})$. Given the bigram assumption for the probability of an individual word, we can compute the probability of a complete word sequence

$$p(w_n|w_1^{n-1}) \approx p(w_n|w_{n-1})$$
$$p(w_1^n) \approx \prod_{k=1}^{n} p(w_k|w_{k-1})$$

This is all we need to go ahead and define document simplicial complex.

4.2 Document simplicial complex

With all the previous sections and chapters we covered the related background work. Now with all this, we can go ahead and define our model to represent a document as a mathematical model and geometrical structure. Before that first, let's see how to catch a word or phrase as a mathematical entity.

As we already discussed that word2vec is the word embedding we will be using to get the semantically meaningful transformation of a word to vector space, we can also use some other word embeddings as [12, 13] and others. This method heavily relies on word embedding. Now before digging deeper let's get an overview of our method stepwise and let's see what are the different dots that we need to connect to reach our goal.

Document Simplicial Complex (DSC), is a variant of an abstract simplicial complex (ASC) or more specifically a somewhat transformed form of ASC to capture the mapping of text to a mathematical form. It is defined on the top of simplex for which the word vectors are the basic building blocks. Let's break down our task and analyze the smaller jobs that we need to get done;

1. Get a one-to-one relation for each word with a word vector in Euclidean space.

2. Catch the phrases and important words while maintaining the order of text in the document.

3. Get the simplex out of the text in the document.

4. Form the collection of simplices as document simplicial complex.

We have already covered the first step and have a solution as word embedding word2vec [10, 11]. Also, to get going with the second step we have the basics covered when we discussed the n-gram model 4.1. Now in upcoming sections and our discussion will be on the methodology to capture the linkage between the n-gram model and phrases selection and also how to get a simplex from a text.

4.2.1 Text Simplex

Given a text document our task is to capture the sentences and phrases from it so that we can incorporate these in out simplicial complex to get the overall structure of a document. Moreover, the phrases and the word collection define the local behavior of text at the paragraph level and sentence level.

n-gram is the contiguous sequence of words from given text, if we capture the same from each sentence for different n as $n \in \{1, 2, 3, 4\}$, we get unigrams, bigrams, trigrams and quadgram and so on. Here our motto is *to capture the role of words and phrases when we decide the semantic meaning of full document.*

For each possible $n \in \mathbb{N}$ for a given text we can get the n-gram from the text, but if we get at the intuitive way of meaningful phrases it naturally comes up to 4 only. So, in our model also let's get till 3-simplex only. Phrases and words will be captures as n-gram which in turns behave as simplex in our method. Each contiguous sequence of n-words is an $(n-1)$-simplex.

Definition 8 (text simplex)**.** *Let the given word sequence be $\{w_0, \ldots, w_k\}$, the n-text simplex $(n \leq k)$ is defined as the n-gram from given sequence.*

Above definition 8, of text simplex makes us able to capture the ordering of words in given text document. Also, the word2vec captures the semantic meaning, hence we have a structure which is capable to do two things; 1) secure the ordering, and 2) seize the semantic meaning.

In this way, text simplex captures all the phrases and words of a text document, where each word is a 0-text simplex and phrases get the higher value of $n \geq 0$ in n-text simplex. To get the importance of each phrase in a text or n-text simplex in the geometrical form of the document we find the frequency of such occurrences and normalize it over all documents. We will discuss all this in detail in the chapter of "experiments and evaluation".

Our method can be divided into four small task that we described earlier, from which the recent development shows the completion of first three tasks. Now we can define our last and main task of defining "Document Simplicial Complex (DSC)".

First, let's get to the geometric analog of a text document, then we will define an abstract simplicial complex in later part of the thesis.

4.2.2 Geometrical analog of DSC

In this section, we will cover the visualization of a document as a geometrical structure and will analyze its properties, the insights that one can get by looking at these entities.

First, let's decide our aim that what is it that one must get from such a visualization. So, in our method we focus on 1) Visualization of a text document, 2) Semantically meaningful behavior, and 3) Connection among various phrases and words. Once we get the visuals of geometrical document simplicial complex, which is the first task, the analysis of those graphs complete the other tasks.

Before putting the geometric DSC's construction formally, let's discuss it in loose words to understand it better. Till now, we have the semantic meaning preserving words, semantic and order maintaining simplices. Now to construct a simplicial complex, collect all n-text simplex and join them in an encouraging way to get our work done. Joining must be in such a way which also captures the importance of that phrase and also shows the various word relationship in a simple manner. We can achieve this by finding "all distinct words occurring in text and joining two words with an edge only if there is a bigram relation, which in chain manner captures the higher n-gram relations".

Now let's define the construction of a DSC as a formal mathematical procedure; Let's fix some notations first, G_n shows the set[1] of n-text simplices and W_n is the corresponding set of counts of n-text simplices.

DSC1 Collect all the 0-text simplices from document and form set G_0 and W_0.

DSC2 Similarly form G_n and W_n for $n \in \{1, 2, 3, 4\}$[2].

DSC3 Each 0-text simplex is a node which is connected with other nodes with a weighted edge if and only if there exists 1-text simplex. Where the weight of an edge is the count of the bigram.

This procedure explains how we capture the relation between different n-simplices as well. If there exists an n-text simplex then from our construction of set G_k with $k \leq n$ ensures that it contains all relevant k-text simplices.

By this methodology one can construct a geometrical structure corresponding to given text, now analyze some graphs to observe if they serve their purpose. If this

[1] Set is the collection well-defined distinct objects.

[2] Up to quad text simplex plays a role in document behaviour above it we do not consider to avoid extended relation and calculation.

method answers; Can we rely on this process to find the similar document? Will it capture the relation between phrases and words?

All the graphs are generated from the tweet collection from twitter [14]. To generate the graphs we formed several small text document by putting various positive, negative and mixed tweets. We removed all the stop words [15] and then formed proposed n-text simplices.

Figures 4.1, 4.2, 4.3 and 4.4 clearly shows the semantically meaningful behaviour of documents. Also, the clusters formed are showing the success of our method that it is able to pull off the task to capture the strong connections between phrases through simplices.

We conclude this chapter with our assignment complete and now dive in for another task to define the abstract analog of document simplicial complex which is a mathematical form of the document. What do we mean by an abstract document simplicial complex? How to compare to such entities? What is the metric or similarity measure to find the similar document by using abstract DSC? These questions will be answered in next chapter.

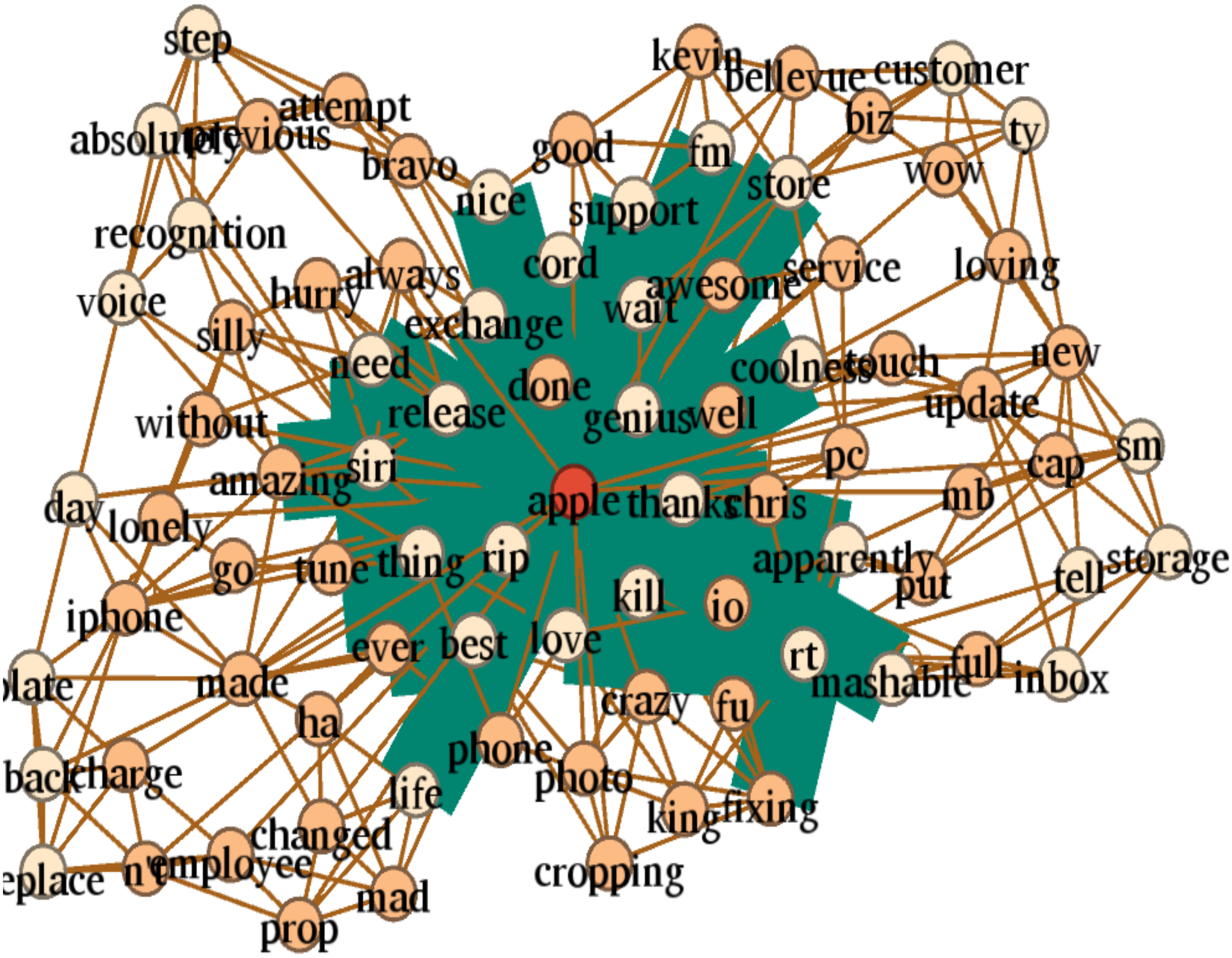

Figure 4.1: This is a graph generated by collection of some positive tweets about products from 'apple'. So if we see from graph then it is evident that word 'apple' is in centre of graph as it is the word, document most talked about, and these green fat edges showing the most used phrase as 'amazing apple 'love apple' etc.

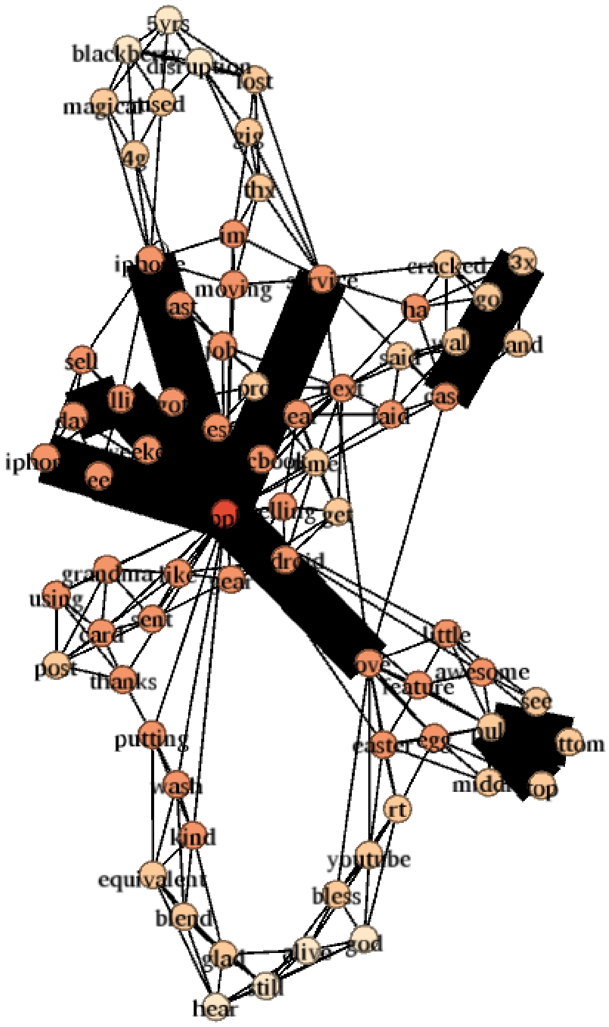

Figure 4.2: Generated from the tweets only about apple, but mixed with some positive and negative comment. Here again, we can see that word 'apple' is in center of the graph and with other words as 'service', 'iPhone' etc. it is showing strong connections. and some other phrases about the features of a product also showing the connection as 'pull button' 'gase 3x' etc.

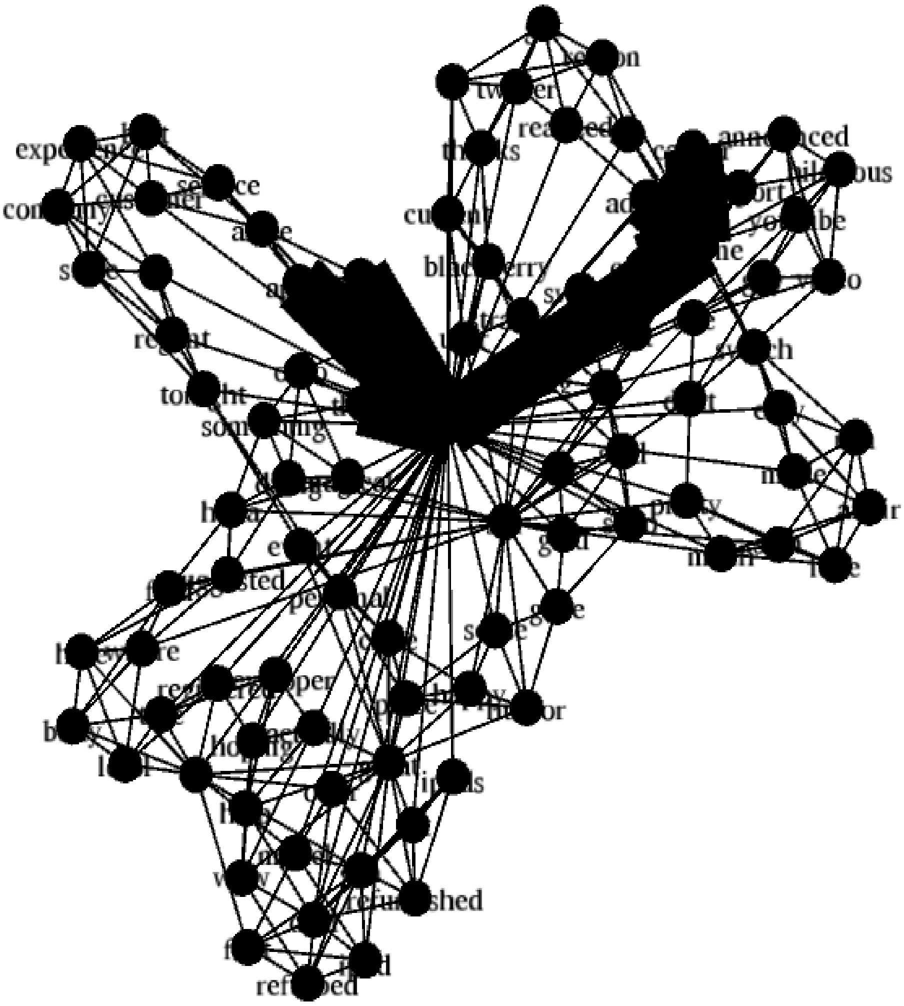

Figure 4.3: One more graph generated from tweets.

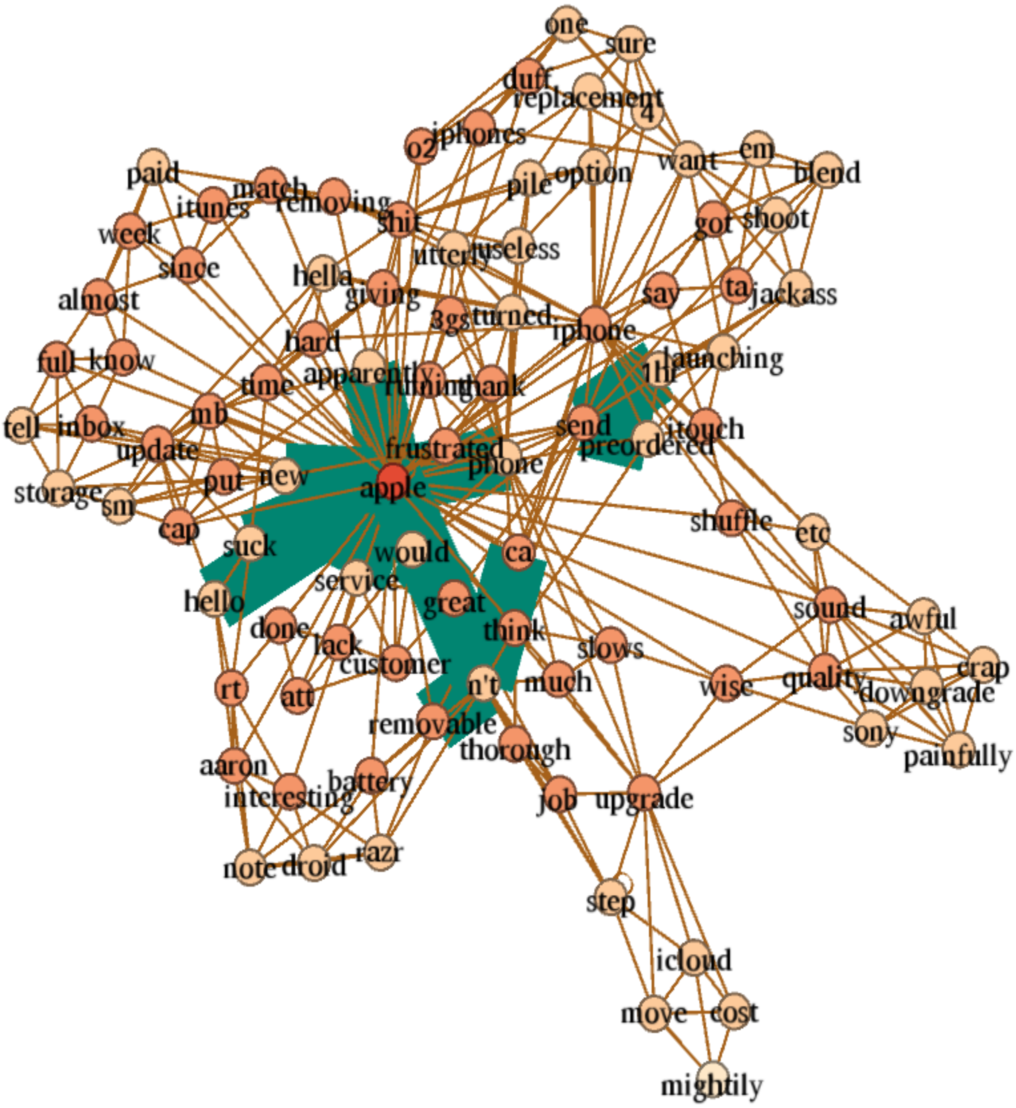

Figure 4.4: Observe the word clusters. in right down part of the graph we see that the words as 'painfully', 'downgrade', 'awful' etc are in one cluster and forming the simplex with word apple, it shows that these are used as phrases in the document. 'apple sound quality awful' forming a 3-simplex here.

Chapter 5

Similarity Metric

Document simplicial Complex (DSC) is a geometric structure which can be seen as a transformation of text to a mathematical form that preserves the semantic meaning and order of text in given document. We already discussed and discovered the DSC construction procedure in the last chapter. Now, what we left with is how to measure the similarity of two given DSC's. Rather comparing the geometrical structure and making things more complex than we prefer the abstract DSC to compare two document for semantic similarity.

The similarity metric is a measure which given two inputs as the simplicial complex, outputs a real value which gives us the information about the resemblance or similarity of given inputs. For us, the input is the document simplicial complex which in turns is a collection of text simplices. Now the similarity metric has to be defined in such a way that it captures the order of text simplices and also preserves the semantic meaning when comparing two simplex or simplicial complex. In this chapter, we will define various methods to do so and also compare the results by evaluating for kNN classification.

Given two probability distribution, earth mover's distance (EMD) is a measure of work done to transform one distribution completely into another one. It also *extends the notion of a distance between the points to that of a distance between sets.* This is the property that we also require in our model to capture the distance between two sets of text simplicies in addition to measure the work done to transform one set of text simplices into other.

EMD finds the distance between two sets given the pairwise distance of elements in both sets. So, we want to have a distance notion between two text simplices. For this purpose we will use Hausdorff distance, set distance and we also see some proposed methods.

5.1 Earth Mover's Distance

Earth mover's distance is a metric to find the similarity between two multi-dimensional distributions in some feature space where the distance between single features is given. The distance between single features is called the *ground distance*. The EMD extends the notion between two single features to that of two sets of distributions, or if we put it in the informal way it "lifts" the distance from single features to full distributions.

To understand it more intuitively, consider in a feature space, one distribution as a mass of earth and other as a collection of holes in that same space. Then the EMD measures the least amount of work needed to fill the holes with earth. Here, a unit of work corresponds to transporting a unit of the earth by a unit of *ground distance*. This can be seen as transportation problem with supply demand.

This can be formalized as the following linear programming problem; Let $P = \{(p_1, w_{p_1}), \ldots, (p_m, w_{p_m})\}$ be the first distribution with m clusters, where p_i is the cluster representative (cluster mean or mode) and w_{p_i} is the weight of the cluster, $Q = \{(q_1, w_{q_1}), \ldots, (q_n, w_{q_n})\}$ be the second distribution with n clusters. Also, $D = [d_{ij}]$ be the ground distance matrix where d_{ij} is the ground distance between clusters p_i and q_j.

We want to find a flow $F = [f_{ij}]$, with f_{ij} the flow between p_i and q_j, that minimizes the overall cost

$$WORD(P, Q, F) = \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} d_{ij},$$

subject to the following constraints:

$$\begin{aligned}
f_i j \geq 0 \quad & 1 \leq i \leq m, \; 1 \leq j \leq n \\
\sum_{j=1}^{n} f_i j \leq w_{p_i} \quad & 1 \leq i \leq m \\
\sum_{i=1}^{m} f_{ij} \leq w_{q_j} \quad & 1 \leq j \leq n \\
\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} = min(\sum_{i=1}^{m} w_{p_i}, & \sum_{j=1}^{n} w_{q_j}),
\end{aligned}$$

The first constraint allows movement from P to Q and not vice versa. The next two constraints limit the amount of supply that can be there from P to their weights, and the cluster in Q to receive no more supplies than their weights. The last constraint forces to move the maximum amount of supplies possible. Once the linear optimization problem gives the solution as optimal flow F, the earth mover's distance is defined as the work normalized by the total flow:

$$EMD(P,Q) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}d_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}}$$

EMD has already been found working well with text [7] and images [16]. In our method, we will implement it for getting the transformation measure as one text simplex set to convert to other.

5.2 Ground distance

From our discussion of EMD in the previous section 5.1 we got to know that to apply it over the two distributions we need the distance between pairwise single features of respective distributions. In our case, to apply EMD for two sets of text simplices, the distance between two simplices from different sets must be known beforehand, which we call the ground distance.

In this section let's discuss what are some possibilities for the ground distance metric. It should be following some rules before we can apply it to use and that are 1) it must preserve the semantic meaning and 2) it should capture the order. We will explore some metrics for our purpose.

Text simplex is a collection of word and we convert each word to the corresponding word vector using word embedding [10], that converts the set of words into a set of vectors in high dimensional feature space. Now one can apply mathematical formulations over such sets. We can suggest ground metric considering word vectors as features. In word2vec model we use 300-dimensional feature space for word vectors.

Definition 8 implies that text simplex is a set of words, now as it has been discussed the earth mover's distance is a notion of distance between sets. So, we can use EMD itself here as the ground distance because we have a single feature as a set only. Now, let's analyze what are some pros and cons of doing so. If one uses the EMD for the ground metric to compute the distance between two text simplices, one also needs to define the distance measure for a lower level feature and that is word vectors. Which puts us in the same situation again to define the ground metric for feature vectors. But, now our single feature is a 300-dimensional vector not a set or distribution, hence we can use Euclidean metric[1]. This solves our one problem to get the ground metric for text simplices. This discussion itself shows how lengthy and expensive model it would be if we used ground metric as EMD for simplices. As it is clear from the definition of EMD in section 5.1, it is an optimization problem and it can be proved that it is a costly affair that suffers from high complexity of $O(N^3 \log N)$. So, using it as a ground metric which works on top of Euclidean distance measure would not be efficient.

Now, let's keep it simple and analyze the situation if we use the ground metric as "set distance". First, let's define the set distance. Given two sets C and D in Euclidean space $\mathbb{R}^N$ the set distance between C and D is defined as

$$setdist(C, D) = \inf\{d(x, y) \mid x \in C,\ y \in D\},$$

where d is Euclidean metric[1]. We defined this set distance for a set of text simplices when we got the text simplices from a given text document, which makes it certain that we will have the only finite set i.e. set of text simplices will be a finite set. This makes our task easy, now inf is min. So for us

$$setdist(C, D) = \min\{d(x, y) \mid x \in C,\ y \in D\}.$$

Set distance will preserve the semantic meaning as we are using the word embedding word2vec but it looses the ordered structure from the text simplices as it will consider a simplex as a set, not as a sequence. On the other hand, the simple procedure to compute the set distance makes it fast to compute and compensate for expensive EMD on the higher level.

Hausdorff distance is also an option for calculating the ground distnce between given two sets of text simplices. It measures the extent to which each point of a set lies near some point of another set i.e. two sets are close in the Hausdorff distance if every point of either set is close to some point of the other set. To put it formally consider two point sets $A = \{a_1, \ldots, a_n\}$ and $B = \{b_1, \ldots, b_m\}$, the Hausdorff distance

[1] If $p = (p_1, \ldots, p_N)$ and $q = (q_1, \ldots, q_N)$ be two points in $\mathbb{R}^N$, the distance would be

$$d(p, q) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (p_N - q_N)^2} = \sqrt{\sum_{i=1}^{N} (q_i - p_i)^2}.$$

is defined as:

$$H(A,B) = max(h(A,B), h(B,A))$$

where

$$h(A,B) = \max_{a\in A}\min_{b\in B} d(a,b)$$

and d is the Euclidean metric. The function $h(A,B)$ is called the *directed Hausdorff distance* from A to B. Hausdorff distance has been extensively used in image processing and computer vision [17, 18].

Let's consider some other distance measure which are not necessarily a metric[2] but we define them in an intuitive way to capture the semantic meaning and also the order of the text simplex as a sequence.

Word2Vec word embedding clusters the similar word, so in the feature space, two similar words are near to each other with very small distance. Now if we have two simplices from near clusters, then if we incorporate only the simplex distance as we discussed some metric above we will lose the positional information of that simplex in feature space. Thus to get it right we will add the centroid theory in our distance measures. Where centroid is the mean of word vectors i.e. given sequence of word vectors $\{v_{w_1}, \dots, v_{w_n}\}$ the centroid of this sequence is $c = \frac{1}{n}\sum_{i=1}^{n} v_{w_i}$. Now the centroid captures the positional information while the simplex word sequence captures the word order of text simplex. So in all discussed distance metrics as a ground distance for EMD, we will add centroid also.

Let's discuss some other metrices now, consider two n-text simplices $s = \{s_1, \dots, s_n\}$ and $t = \{t_1, \dots, t_n\}$, where s_i and t_j denote the word vecotrs for ith and jth word in text simplex s and t respectively. Also let's denote the centroid of two text simplices as c_s and c_t respectively. Then we define the distance between two text simplices as;

$$dist(s,t) = \sum_{i=1}^{n} d(s_i, t_i) + d(c_s, c_t)$$

where d is the Euclidean metric[1]

[2] A metric in a set X is a function $f : X \times X \to [0, \infty)$ which satisfy all the following conditions for all $x, y, z \in X$; (1) $f(x,y) \geq 0$, (2) $f(x,y) = 0$ iff $x = y$, (3) $f(x,y) = f(y,x)$ and (4) $f(x,z) \leq f(x,y) + f(y,z)$.

Chapter 6

Evaluation

Previous chapter 5 covered the metrics that we need to compare two simplices, these distance measures will come handy once we go for the evaluation of our method and compare it with other methods. We also discussed in chapter 2 some relevant methods that we need to compare with which includes WMD, LDA, LSI etc.

In this chapter, we will first discuss some benchmark datasets for text classification tasks. We will describe total seven datasets for text categorization in the context of kNN classification. Then we will compare the nearest neighbor performance of the proposed method and the competing methods on these datasets.

Document similarities are particularly useful for classification given a supervised training dataset, via the kNN decision rule [19]. kNN provides an interpretable result in form of nearest neighbor that allows one to verify the prediction results. So let's first discuss kNN in brief that how does it work? and what is kNN?

kNN is a non-parametric, lazy learning algorithm. Its purpose is to use a database in which the data points are separated into several classes to predict the classification of a new sample point. Non-parametric means that it does not make any assumptions on the underlying data distribution. which is pretty useful for "real world" datasets. Also, the term "lazy algorithm" here means that it does not use the training data points to do any generalization. In other words, there is no explicit training phase or it is very minimal.

kNN algorithm is based on feature similarity: How closely out-of-sample features resemble our training set determines how we classify a given data point. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors. The small value of k overfits the boundary while large value makes it go underfit. Value selection for k is usually depended on the dataset.

Table 6.1: Dataset statistics.

Name	c	Train	Test	s	BOW Dim.	Unique Grams (Avg.) Uni	Bi	Tri
20News	20	11293	7528	14.48	29671	93.64	113.11	103.60
Reuters	8	5485	2189	7.45	22425	36.11	49.05	51.39
Ohsumed	10	3999	5153	8.2	31789	64.05	78.5	76.30
Classic	15	4985	2128	4.67	24277	41.83	48.77	46.73
BBCsport	5	517	220	17.09	13243	126.10	151.26	138.28
Amazon	4	5600	2400	7.6	42063	54.09	58.18	51.58

6.1 Data Sets and Preprocessing

In this section, we describe all seven benchmark datasets that we used to compare our method with existing classic and state-of-the-art methods. The dataset collection is based on sentiment analysis and also the categorical data. Some of the datasets are multi labeled also i.e. some of the entities in the dataset have more than one label.

20News[1]: It is a collection of approximately 20,000 newsgroup documents. The data is categorized into 20 different topics, in which some of the topics are closely related. We use dataset sorted by date.

BBC Sport: Collection of news article from BBC sport website corresponding to sports news in five topical areas from 2004-2005. It is categorized in *'athletics', 'cricket', 'football', 'rugby', 'tennis'.*

Classic4: It contains set of sentences from academic papers labeled by publisher name. The dataset contains four categories *'cran', 'cisi', 'casm', 'med'*, each category set contains over 1000 documents.

Ohsumed**: A collection of medical abstracts of the year 1991, categorized in 23 cardiovascular disease groups. We consider the small subset consisting of the 10 classes.

Reuters8**: Corpus consists of news stories appeared on Reuters newswire in 1987 categorized in 135 classes. We consider dataset consisting of only 8 classes Reuters8.

Twitter [14]: Tweets labeled with sentiments *'positive', 'negative', 'neutral'.*

Amazon: Product reviews labeled by product category *'book', 'dvd', 'kitchen', 'electronics'.* This dataset is in contrast with sentiment analysis.

[1] `http://qwone.com/ jason/20Newsgroups/`

** Corpus is multilabel

Word2vec model, which has embedding for 3 million words/phrases, has been used to get embedding for words, to construct our DSCs. Word2Vec model is freely available[2] to use and it is trained [11] on Google news. We restricted our uni-grams for each document only to the vocabulary of embedding model i.e. we discarded any word which has no embedded representation in word2vec model, but for all baseline methods, we used the document as a whole i.e. no dropping of words. We use SMART stop list [15] to remove all insignificant words from document.

For each benchmark dataset that we use to compete with baseline methods, table 6.1 reports relevant statistics including average number of unique uni-grams, bi-grams and tri-grams per document, average number of sentences s per document, number of classes c in corpus, where *train* and *test* shows number of inputs that we use as building set and search set respectively.

For the dataset with given split of training and testing subsets, we use them as it is for building and testing set in kNN classification task and for the datasets AMAZON, CLASSIC, BBCSPORT and TWITTER with no pre-defined train/test split we use 70-30 split for classification purpose.

Chapter 2 discusses and give an overview of some methods and metrics that are used for comparing our method. For each baseline method, we use the Euclidean distance for kNN classification. All the parameters are set on heuristic based and can be found by Bayesian optimization for all algorithms [20].

6.1.1 Chunks in a document

In chapter 4, the construction of document simplicial complex has been discussed. We observe the stepwise procedure to get DSC from a given text document, we used word embedding to get a mapping of each word in high dimensional Euclidean space. If we capture the whole document as one complex we then lose the topic level information. For example; a research paper is normally written in specific order as abstract, introduction, related work, methodology and the results.

It shows that a document information is spread out in small chunks within a document. Now to capture chunk wise information of document we consider a text document as a collection of small subdocuments i.e. chunks. So, while constructing the DSC we build a complex for each chunk and the final DSC for given document is the collection of subcomplexes constructed from chunks.

Such construction gives us an upper hand to dig more into the document and get

[2] `https://code.google.com/p/word2vec/`

to know a text more. This explores the topic wise classification task for a document. Chunk level complex comparison can give us information about an unknown document to closely categorize it with known documents.

In datasets with respectively less number of sentences or paragraphs per document such as AMAZON and TWITTER, to capture the section wise behavior of a document we use the low value of chunks. While calculating the flow of a document we use less than 30 unique words to capture the similar words across the document, it is also efficient to use.

6.1.2 Distance Measure for DSC

We have discussed some possible metrics for our task to find the distance between two given simplices. In chapter 5 we explored some of the possibilities. Hausdorff and set distance are the two main competitors for this as both are efficient to use and also are metric in mathematical feature space. Before concluding the chapter we also constructed one distance measure, in this chapter we will see how it helps to categorize the document.

Given the document we can construct simplices and can also compare two simplexes, the collection of simplex will give us a DSC which is the proposed representation for a document. In this section, we will see how to compare two DSC's. A document simplicial complex is collection of various n-text simplices where we restrict n to the set $\{1, 2, 3, 4\}$. Using a metric discussed in chapter 5 we can get the pairwise distance for given two sets of k-text simplices and hence we have a distance matrix. This matrix works as a ground distance for comparing two sets of text simplices when using EMD.

Once we have the distance measure for respective pair of text simplex sets from two documents we use them to get the distance between two document complexes. We can use each text simplex set distance also to represent the DSC distance or can collect the information from each set and accumulate it for DSC. Consider for two document complex X and Y we have the respective k of the n-text simplex sets as $\{G_{1,X}, \ldots, G_{k,X}\}$ and $\{G_{2,Y}, \ldots, G_{k,Y}\}$. Using the metric defined in chapter 5 and EMD on top of that we can get the distance between $G_{i,X}$ and $G_{i.Y}$ for $i \in \{1, \ldots .k\}$ as $DM_{i,X,Y}$.

Some possible options for distance measure between two DSC's are as follows; 1) we can consider one of the $DM_{i,X,Y}$ as distance, 2) we can take sum of all the distance measures, $sumDist(X, Y) = \sum_{i=1} DM_{i,X,Y}$

3) average of all the measures, $avgDist(X,Y) = \frac{1}{k}\sum_{i=1}^{k} DM_{i,X,Y}$,

4) unigrams have the positional knowledge of a text document while higher grams focus on order in given document, so we can have

$$Dist(X,Y) = \alpha DM_{i,X,Y} + \beta \sum_{i=2} DM_{i,X,Y}$$

in this equation if we have $\alpha > \beta$ we have *uniDist*, else we define it as *ordDist* which focus on order of words in document.

5) Euclidean norm of all the distances, $normDist(X,Y) = \sqrt{\sum_{i=1}^{k} DM_{i,X,Y}^2}$

6.2 Results

KNN decision rule provides an interpretable result in the form of nearest neighbors. We get the similarity based on kNN rule which can be used to compare for ranking and recommendation. To prove the efficiency of the proposed method on classification we compare the kNN results of our model with each aforementioned document representation and distances. We use the neighborhood size of kNN from $k \in \{1, \ldots, 19\}$

Figure 6.1 shows the comparative result on the stacked bar graph for Hausdorff and set distances used as the ground distance in EMD for different k. This is from $20NewsGroup$ dataset when we used only 100 files for training and 25 files for testing the method.

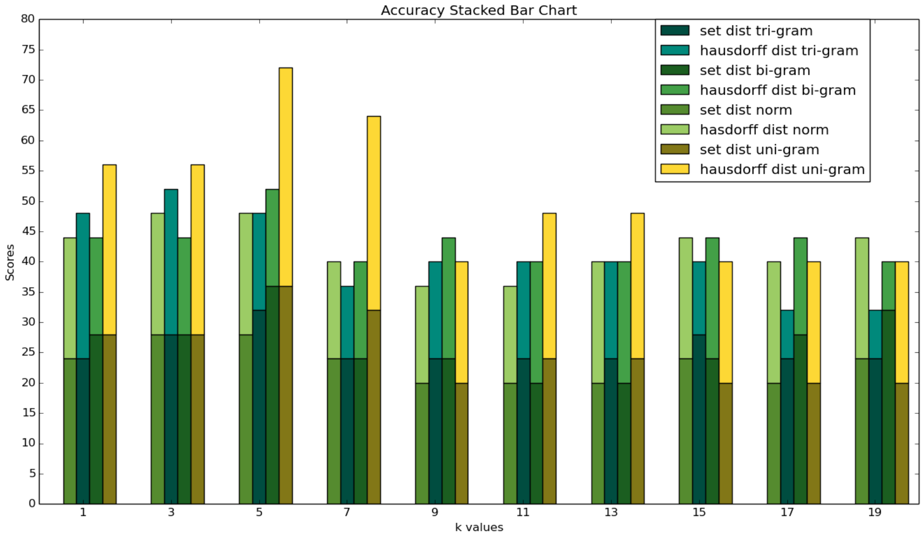

Figure 6.1: stacked

In the following figure 6.2 it is shown that how much accuracy we are getting over WMD when we used different metrics that we proposed for comparing two DSC's and we got the result on two datasets as twitter and 20News. We did not use the whole dataset as it was taking too much time so we restricted our task up to 1000 files.

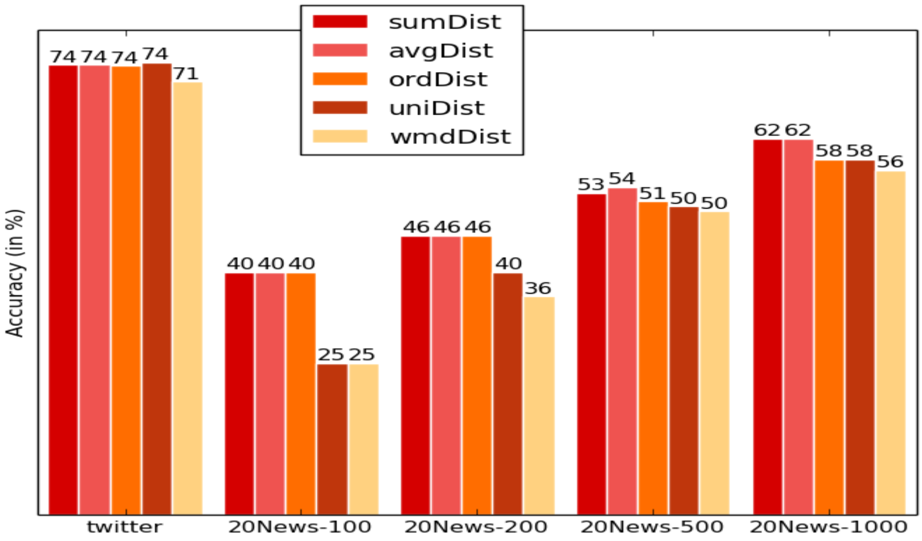

Figure 6.2: main result

Figure 6.3 shows the same result in graphical manner rather in histograms.

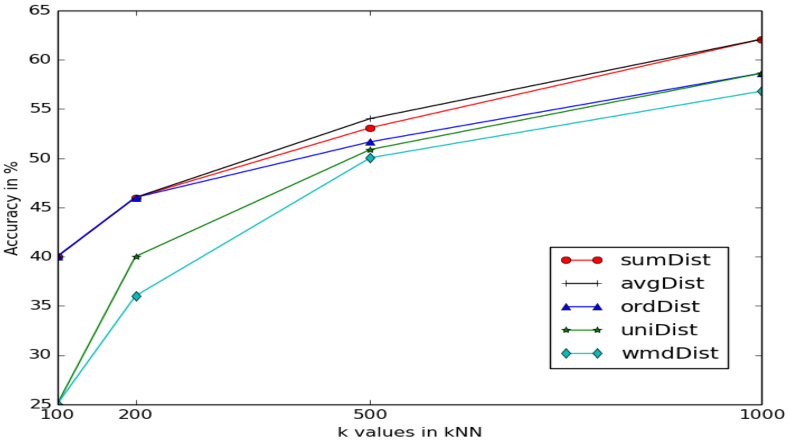

Figure 6.3: increase file size comparison

Figure 6.4 shows the result for twitter dataset at different value of k in kNN.

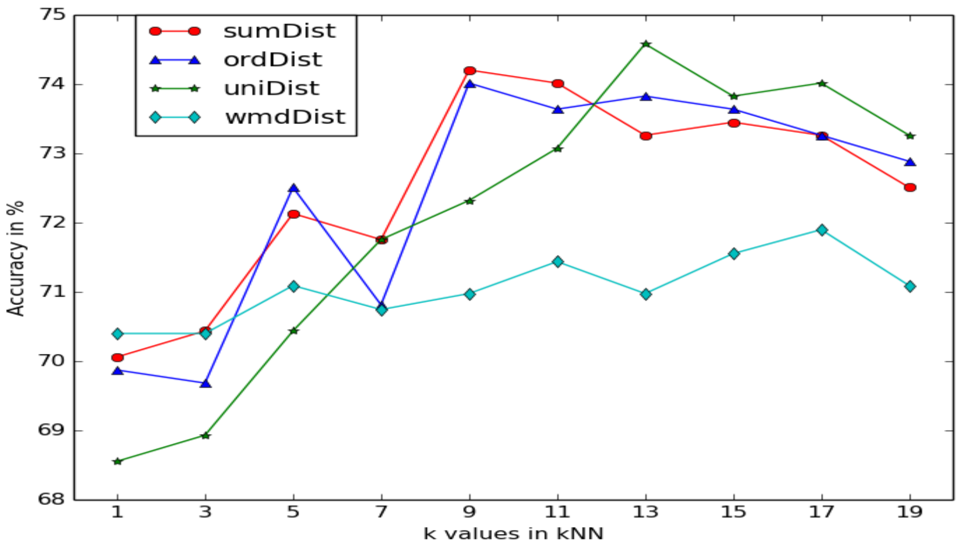

Figure 6.4: twitter comparison

6.3 Scope of future work and conclusion

Document simplicial complex is a representation which is leading us to get low error rates compared to baseline methods. The reason behind such results can be the several level wise features capturing and word embedding. We observed a document on different levels as n-grams, topic level and subdocument and this make our method better.

As we compared our method with several baseline methods we observe that we are getting 4-5% accuracy over state-of-the-art methods. In our present proposed work, we have to work on to make it faster as currently it is very slow compared to other methods and also we need to observe some other possibilities to compare two documents as to how can we compare on sentence level? can we embed the grammar in the proposed method?

We would also like to explore the idea of persistent homology in our representation. The main idea is to look the document as in two parts horizontal and vertical where a horizontal part is sentence and topic level visualization of the document and vertical would be to consider the birth and death of several n-grams

References

[1] S. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3 109–126.

[2] D. M. Blei, A. Y. Ng, M. I. Jordan, and J. Lafferty. Latent dirichlet allocation. *Journal of Machine Learning Research* 3, (2003) 993–1022.

[3] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE* 41, (1990) 391–407.

[4] X. Glorot, A. Bordes, and Y. Bengio. Domain Adaption for large scale sentiment classification: A deep learning approach. In Proceedings of the 28th International Conference on Machine Learning. 2011 513–520.

[5] M. Chen, Z. E. Xu, K. Q. Weinberger, and F. Sha. Marginalized Denoising Autoencoders for Domain Adaptation. In ICML. icml.cc / Omnipress, 2012 .

[6] Wikipedia. Latent Dirichlet Allocation.

[7] M. J. Kusner, Y. Sun, N. I. Kolkin, and K. Q. Weinberger. From Word Embeddings To Document Distances. In ICML, volume 37 of *JMLR Workshop and Conference Proceedings*. JMLR.org, 2015 957–966.

[8] J. Munkres. Elements of Algebraic Topology. WestView Press, 1996.

[9] Wikipedia. Abstract Simplicial Complex.

[10] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representation in vector space. In Proceedings of Workshop as ICLR. 2013a .

[11] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representation of words and phrases and their compositionality. In NIPS. 2013b 3111–3119.

[12] J. Pennington, R. Socher, and C. D. Manning. GloVe: Global Vectors for Word Representation. In Empirical Methods in Natural Language Processing (EMNLP). 2014 1532–1543.

[13] O. Barkan. Bayesain Neural Word Embedding. In ssociation for the Advancement of Artificial Intelligence (AAAI). 2017 .

[14] N. Sanders. Sanders-twitter sentiment corpus 2011.

[15] G. Salton and C. Buckley. The smart retrieval systemexperimets in automatic document processing 1971.

[16] Y. Rubner, C. Tomasi, and L. J. Guibas. A metric for distributions with applications to image databases. In IEEE International Conference on Computer Vision (ICCV). 1998 59–66.

[17] D. Huttenlocher, G. Klanderman, and W. Rucklidge. Comparing images using the Hausdorff distance. In IEEE Transactions on Pattern Analysis and Machine Intelligence. 1993 850–863.

[18] B. Sendov. Hausdorff distance and image processing. *Russian Mathematical Surveys* 59, (2004) 319.

[19] T. Covert and P. Hart. Nearest neighbor pattern classification. In Information theory, IEEE Transactions. 1967 21–27.

[20] J. Snoek, H. Larochelle, and R. P. Adams. Practical Bayesian optimization of machine learning algorithms. In Neural Information Processing Sysytems (NIPS). 2012 2951–2959.