A Review of the applications of n-gram graphs

Ayush Pareek

March 23, 2017

1 Introduction

In the domain of natural language processing, there have been a number of methods using n-grams. An n-gram is a, possibly ordered, set of words or characters, containing n elements.

Examples of n-grams from the sentence: My name is Ayush.
Word unigrams: My, name, is, Ayush
Word bigrams: My name, name is, is Ayush
Character bigrams: My, y , n, na , am, ...
Character 4-grams: My , y na , nam , name...

1.1 N-gram Graphs

Graphs have been used to determine salient parts of text [1,2,3] or query related sentences [4]. Lexical relationships [23] or rhetorical structure [5] and even non-apparent information [6] have been represented with graphs. Graphs have also been used to detect differences and similarities between source texts [7], inter-document relations [8], as well as relations of varying granularity from cross-word to cross-document, as described in Cross-Document Structure Theory [9]).

1.2 Features of N-gram Graphs

Some general properties of n-gram graphs which make them suitable for applications related to Document Understanding and Processing are-

1. **Purely Statistical:** method which offers richer information than widely used representations such as the vector space model.

2. **Preprocessing Free:** preprocessing step used in most Text processing algorithms usually involves stemming and stop-word removal among other non-efficient language dependant operations.

3. **Language Neutral:** A method that does not require language dependent resources (thesauri, lexica, etc.) can be applied directly to different languages which are not rich in text mining resources and do not have much research done on them.

4. **Extracts and retains neighbourhood information:** The neighbourhood information between different linguistic units is a very important factor for determining the meaning of these units. This quality is integrated in n-gram graphs using weighted edges.

5. **Nth order Collocation Information (Neighbour-of-a-Neighbour):** The graph in itself is a structure that maintains information about the 'neighbour-of-a-neighbour'. This means that if A is related to B through an edge or a path in the graph and B is related to C through another edge or path, then if the proximity relation is considered transitive we can deduce that A is related to C.

6. **Retains Sequence Information:** The edge weights could be determined such that they preserve sequence information. Example; Using Non-symmetric approach described next.

1.3 Neighbourhood Types

1. **Non-symmetric approach** - retains sequence information. Each edge is weighted by the number of co-occurrences of the neighbours within a given window of the text.

2. **Symmetric approach** - indicates collocation. Each edge is weighted based on the number of co-occurrences of the neighbours within a window in the text.

3. **The Gauss-normalized symmetric approach** - retains distance information. Each edge is weighted based on the number of co-occurrences of the neighbours within the text and the neighbours' distance at each occurrence.

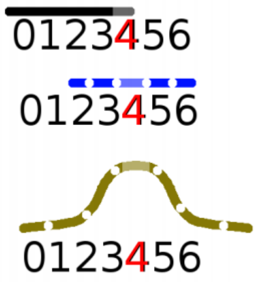

Figure 1: Different types of n-gram windows (top to bottom): non-symmetric, symmetric and Gauss-normalized symmetric. N-gram 4 is the n-gram of interest.

1.4 Major Applications

1. Summarization Tasks
2. Text Classification
3. Sentiment Analysis
4. Fuzzy string matching
5. Record Linkage
6. Authorship Identification
7. Multi-Class Text Classification
8. Semantic Annotation
9. Text Stemmology
10. Entity Name Service

1.5 Major Graph Operators and techniques useful in NLP

Given two instances of n-gram graph representation G1, G2, there is a number of operators that can be applied on G1, G2 to provide the n-gram graph equivalent of union, intersection and other such operators of set theory. Moreover, n-gram graphs can be made updatable i.e. a model that could easily change when a new document enters the set. Besides, techniques have been developed to identify and remove the non-interesting parts of data that hinder the task i.e. noise.

1. **Graph Similarity and Containment**
2. **Graph Union and Intersection**
3. **Delta and Inverse Intersection**
4. **Updatability**
5. **Removing noise using n-gram graph operators**

1.6 Attributes of n-gram graphs

1. **Indication of nth order relations:** The graph in itself is a structure that maintains information about the 'neighbour-of-a-neighbour'. This means that if A is related to B through an edge or a path in the graph and B is related to C through another edge or path, then if the proximity relation is considered transitive we can deduce that A is related to C. The length of the path between A and C can offer information about this indirect proximity relation. This can be further refined, if the edges have been assigned weights indicating the degree of proximity between the connected

vertices. This attribute of the n-gram graphs primarily differentiates them from the vector space representation. If one creates a feature vector from an n-gram graph, where the edges correspond to dimensions of the feature vector, the indirect relation between vertices is lost. Keeping information on an n-th order relation between a number of distinct elements in a vector will require adding a dimension for every relation between two elements. Therefore, even though one may construct a vector in a way that it will represent the same information as the n-gram graph, there is high complexity in the transformation process. However, further study of the equivalence of the two representations would be interesting to perform in the future.

2. **Parametrically determined generality of the model**The n-gram graph, if viewed as a language model, recognizes sequences of symbols based on the way they are found to be neighbours. The set of sequences the model accepts is actually the solution to a problem of constraint satisfaction [10], where the constraints are based upon the neighbourhood relation

3. **Language neutrality**. When used in Natural Language Processing, the ngram graph representation makes no assumption about the underlying language. This makes the representation fully language-neutral and applicable independent even of writing orientation (left-to-right or right-to-left) when character n-grams are used. Moreover, the fact the the method enters the sub-word level has proved to be useful in all the cases where a word appears in different forms, e.g. due to difference in writing style, inflection of word types and so forth.

4. **Generic application** The n-gram graph need not be applied to text representation only. The existence of small-length symbols, like character n-grams, that are related via a neighbourhood relation can also be found in such applications as image analysis and data mining or bioinformatics. The graph in itself is a mathematical construct with no predefined semantics, suitable for the representation of relations. This generic utility makes it usable whenever there is a meaningful neighbourhood relation. Furthermore, the amount of information contained within a graph is only restricted by the data one desires to annotate graph edges and vertices with.

2 Application in Summarization Tasks

1. **Content Selection:** Given the content definition and the chunking process, each sentence is assigned a score, which is actually the sum of the similarities of its chunks to the content. This process, we call chunk scoring, overs an ordered list of sentences L. Another alternative, we call sentence scoring, would be to assign to each sentence its similarity to the

content, without chunking. Given the sentences' ordered list, a naive selection algorithm would select the highest-scoring sentences from the list, until the summary word count limit is reached. However, this would not take redundancy into account and, thus, this is where redundancy removal comes in.

2. **Query Expansion:** Query expansion is based on the assumption that a set of words related to an original query can be used as part of the query itself to improve the recall and usefulness of the returned results. In the literature much work has indicated that query expansion should be carefully applied in order to improve results [11]

3. **Redundancy Removal:** The redundancy removal process has two aspects-

   (a) **The inter-summary redundancy (user-modeled redundancy):** The inter-summary or user-modeled redundancy refers to the redundancy of information apparent when the summarization process does not take into account information already available to the reader.

   On of the Redundancy removal methods proposed [GIANNAKOPOULOS at el.] via n-gram graphs is as follows:-

      i. Extract the n-gram graph representation of the summary so far, indicated as G_{sum}.

      ii. Keep the part of the summary representation that does not contain the content of the corresponding document set U, $G^{'}_{sum} = G_{sum} \Delta C_U$

      iii. For every candidate sentence in L that has not been already used

         A. extract its n-gram graph representation, G_{cs}

         B. keep only $G^{'}_{cs} = G_{cs} \Delta C_U$, because we expect to judge redundancy for the part of the n-gram graph that is not contained in the common content C_U

         C. assign the similarity between $G^{'}_{cs}; G_{sum'}$ as the sentence redundancy score.

      iv. For all candidate sentences in L, Set the score of the sentence to be its rank based on the similarity to C_U minus the rank based on the redundancy score.

      v. Select the sentence with the highest score as the best option and add it to the summary.

      vi. Repeat the process until the word limit has been reached or no other sentences remain.

   (b) **The intra-summary redundancy:** The intra-summary redundancy refers to the redundancy of a sentence in a summary, given the rest of the content of the summary. In order to ensure intra-summary non-redundancy, one has to make sure that every sentence added only minimally repeats already existing information.

4. **Summary System Evaluation**

- **The AutoSummENG method (Giannakopoulos et al., 2008) (AUTOmatic SUMMary Evaluation based on N-gram Graphs)**

  The AutoSummENG methodology evaluates a set of summarizing systems, or 'peers', with respect to a given set of model summaries. In order to perform this kind of evaluation, the system compares an n-gram graph representation of each peer summary to the n-gram graph representations of the model summaries, grading each system by a set of similarity indications: one similarity indication for each model summary on the same topic.

  The AutoSummENG system can function on either n-grams of words or characters, each giving its own results upon application. It has been shown by means of analysis, that if one uses the toolbox to evaluate how responsive the summaries of a given system are, it is preferable to use character n-grams. The graph of n-grams (whether words or characters) are created by having each n-gram represented by a node in the graph, and add edges between neighbouring n-grams. N-grams are considered neighbours if they fall within a number of words (or characters correspondingly) of each other. The toolkit also takes into account n-grams of different sizes, which can be set manually. Therefore, the parameters the user can test are:

  – Whether character n-grams or word n-grams are to be used.
  – The minimum size of n-grams to take into account.
  – The maximum size of n-grams to take into account.
  – The distance within which two n-grams are to be considered neighbours.
  – Whether for the Value grade, the number of co-occurrences of two ngrams may be taken into account, or the average distance between the neighbouring n-grams in these occurrences.

  The grades (values) returned by the toolkit are supposed to provide a ranking similar to that humans would decide on if they evaluated the same set of peers over the same topics. That ranking, should be expected to be correct, given a sufficient number of topics to evaluate on. There are different kind of grades returned for a single comparison:

  (a) The first refers to overlap, indicating how much the graph representation of a model summary overlaps a given peer summary. This is called the Co-occurrence measure.
  (b) The second also refers to overlap, also taking into account how many times two n-grams are found to be neighbours. This is

called the Value measure, which expects two similar texts to have n-grams neighbouring about the same number of times.

(c) The third is the Size measure, simply indicating the ratio of n-grams between the smaller and larger summary (whether model or peer).

(d) The final grade is the Overall grade, which is a weighted sum of the previous measures and is in a planning stage.

The implementation of the AutoSummENG method allows for the use of multithreading for higher performance.

- **MeMoG: The Merged Model Graph**

In this technique for summary evaluation, the update operator U for the n-gram graphs (similar to the merging operator)is used. It which allows the creation of a “centroid” graph. The update function $U(G_1, G_2, l)$ takes as input two graphs, one that is considered to be the pre-existing graph G_1 and one that is considered to be the new graph G_2. The function also has a parameter called the learning factor $l \in [0, 1]$, which determines the sensitivity of G_1 to the change G_2 brings.

Focusing on the weighting function of the graph, resulting from the application of $U(G_1, G_2, l)$, the higher the value of learning factor, the higher the impact of the new graph to the resulting graph of the update. More precisely, a value of l = 0 indicates that G_1 will completely ignore the (considered new) graph G_2. A value of l = 1 indicates that the weights of the edges of G_1 will be assigned the values of the new graph’s edges’ weights. A value of 0.5 gives us the merging operator. The definition of the weighting performed in the graph resulting from U is:

$$W^i(e) = W_1(e) + (W_2(e) - W_1(e)) \times l$$

The U function allows using graphs to model a whole set of documents: in our case the model set. The model graph creation process comprises the initialization of a graph with the first document of the model set and the updating of that initial graph with the graphs of following model summaries. Especially, when one wants the overall graph’s edges to hold weights averaging the weights of all the individual graphs that have contributed to it, then the i-th new graph that updates the overall graph should use a learning factor of $l = \frac{1}{i}$, $i > 1$. This gives a graph that has a role similar to the centroid of a set of vectors: it functions as a representative graph for the set its constituent graphs.

- **NewSum: “N-Gram Graph”-Based Summarization in the Real World**

  NewSum is an n-gram graphs based technique used for multilingual multi-document news summarization. The system uses the representation of n-gram graphs in a novel manner to perform sentence selection and redundancy removal for the summaries and faces problems related to topic and subtopic detection (via clustering) and multilingual applicability, which are caused by the nature of the real-world news summarization sources. It doesn’t require training and can be used as an everyday tool. It can summarize news from a variety of sources, using language agnostic methods.

2.1 Advantages

Some salient benefits of using n-gram graphs in summarization are-

1. **Free of Preprocessing and Language Neutrality:** These methods do not require language-dependent preprocessing or resources (thesauri, lexica, etc.).

2. **Full automation:** The complete process is free of human intervention, apart from the human model summaries.

3. **Context- Sensitivity:** The methods consider contextual information, so that wellformedness of text is taken into account. Wellformedness can be loosely defined as the quality of a text that allows easy reading. A text that is a random sequence of words would lack this quality, even if the words are on topic.

4. **Tolerant to noise:** Based on the task, we can usually identify non-interesting parts of data that hinder the task. This ‘noise’ can be removed via the already proposed n-gram graph algorithms.In the case of a classification task, we create a merged graph for the full set of training documents T_c belonging to each class c. After creating the classes’ graphs, one can determine the maximum common subgraph between classes and remove it to improve the distinction between different classes.

2.2 Shortcomings

- NewSum (method for multilingual multi-document news summarization) faces problems related to topic and subtopic detection (via clustering) and multi-lingual applicability, which are caused by the nature of the real-world news summarization sources.

3 Sentiment Analysis

Sentiment analysis aims to determine the exact polarity of a subjective expression. Specifically in [12] there is an effort using semi-supervised machine learning methods to determine orientation of subjective terms by exploiting information given in glosses provided by WordNet.

The methodology using n-gram graphs involves three steps:

1. **Disambiguation of word senses (WSD)**

   This WSD algorithm performs disambiguation for every word of each headline of our corpus, taking as input a headline and a relatedness measure [13]. Given such a measure, it computes similarity score for word sense pairs, created using every sense of a target word and every sense of its neighbouring words. The score of a sense of a target word is the sum of the maximum individual scores of that sense with the senses of the neighbouring words. The algorithm then assigns the sense with the highest score to the target word. The algorithm supports sev- eral WordNet based similarity measures, and among these, Gloss Vector (GV) performs best for non lit- eral verbs and nouns [14]. In order to measure similarity between two word senses, the cosine similarity of their corresponding gloss vectors is calculated. The input to the algorithm is the corpus enriched with Part-of- Speech (POS) tags performed by the Stanford POS tagger.

2. **Assignment of polarity to word senses, based on the results derived from the WSD step**

   This step detects polarity of the senses computed dur- ing the first step. To do this, WordNet senses associ- ated with words in the corpus are mapped to models of positive or negative polarity. These models are learned by exploiting corresponding examples from the Gen- eral Inquirer (GI). To compute the models of positive and negative polarity and produce mappings of senses to these models, a graph based method based on character n- grams is used, which takes into account contextual (neighbourhood) and sub-word information. The (directed) edges of the graph are labeled by the concatenation of the labels of the vertices they connect in the direction of the connection. The edges $e^G \in E^G$connecting the n-grams indicate proximity of these n-grams in the text within a given window D_{win} of the original text . The edges are weighted by measuring the number of co-occurrences of the vertices' n-grams within the window D_{win}.

   To compute models of polarity using n-gram graphs, two sets of positive and negative examples of words and definitions provided by the General Inquirer (GI) are used.

   To represent a text set using n-gram graphs, an update/merge operator between n-gram graphs of the same rank is used. Specifically, given two graphs, G_1 and G_2, each representing a subset of the set of texts, a single graph that represents the merging of the two text subsets can be created: update(G_1,G_2) $= G_u = (E_u,\ V_u,\ \mathrm{L}, W_u)$, such that $E_u = E_G^1 \cup E_G^2$ where E_G^1, E_G^1 are the edge sets of G_1,G_2 correspondingly.

The weights of the resulting graph's edges are calculated as follows: $W^i(e) = W_1(e) + (W_2(e) - W_1(e)) \times l$. The factor $l \in [0, 1]$ is called the learning factor: the higher the value of learning factor, the higher the impact of the second graph to the first graph. The model construction process for each class (e.g. of the positive/negative polarity class) comprises the initialization of a graph with the first document of a class, and the subsequent update of this initial graph with the graphs of the other documents in the class using the union operator. As we need the model of a class to hold the average weights of all the individual graphs contributing to this model, functioning as a representative graph for the class documents, the i-th graph that updates the class graph (model) uses a learning factor of $l = \frac{i-1}{i}$, $i > 1$. When the model for each class is created, we can determine the class of a test document by computing the similarity of the test document n-gram graph to the models of the classes: the class whose model is the most similar to the test document graph, is the class of the document. More specifically, for every sense x of the test set, the set of its synonyms (synsets) and Gloss Example Sentences (GES) extracted from WordNet, are being used for the construction of the corresponding n-gram graph X for this sense.

3. **Polarity detection on a sentence level, by exploiting polarities of word senses and contextual cues such as valence shifters**

   For the sentence level polarity detection, two HMMs (Hidden Markov Models) [15] are trained - one for the positive, and one for the negative cases. The reason behind the choice of HMMs was that they take under consideration transitions among observations which constitute sequences. The POS of a word combined with the word's polarity constitutes an observation. This information is provided by the POS tagging, and the graph based polarity assignment method upon metaphorical and expanded senses of the input sentences. The transitions among these observations yield the polarity of the sentential sequences. Structured models have been exploited for polarity detection showing promising re- sults [16]. To exploit valence shifters, these are man- ually annotated in the corpus: they are assigned a predefined value depending on whether they revert, strengthen or weaken the polarity, in order to be integrated in the HMM. Having trained two HMM's, one for positive and one for negative cases, the polarity of each headline sentence can be determined by means of the maximum likelihood of the judged observations given by each HMM. In order to evaluate this method of classifying headlines containing metaphors and expanded senses into positive and negative, a 10-fold cross validation method for each of the two subsets can be used.

3.1 Advantages

1. Language Neutral
2. Robust
3. Efficient
4. Tolerant to noise

4 Application in Entity Subscription Service

An entity subscription service informs subscribed users of changes in the descriptive data of an entity, which is a set of attribute name-value pairs. The subscription service is a supplement to the usability of the ENS, tackling the problem of keeping consumers of information concerning specific entities updated. It aims to deliver ranked descriptions of the changes on entities, following user preferences through a feedback-driven adaptation process. The adaptation is based on both the content and the type of each entity change. We evaluate the learning curve of the system and the utility of the content-type discrimination. The method involving n-gram graphs demonstrate good results, especially in the system's content-aware adaptation aspect.

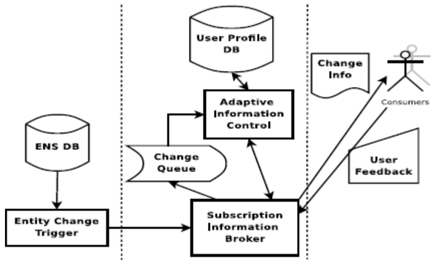

Figure 2: Schematic of the Subscription Service and its interactions

The use of n-gram graphs within this work is due to several of their traits like language neutrality and the fact that, when used for matching between strings, they offer a graded normalized indication of similarity. The updatability [17] of n-gram graphs is another feature which is very useful in this context. In other words, if we judge a set of attribute names-values as indicative of importance, we can create an n-gram graph that models the whole set and, thus, avoid keeping all the attribute names-values for matching. Furthermore, the model offers fuzzy matching and substring matching which helps in open domains of attribute names and values, as is the case of an ENS.

4.1 Representation of Changes

To model the content C of changes a user is interested in, we create for each training instance C_i, given by the feedback process, a corresponding n-gram graph G_{C_i} . The graph is based on the string representation of the change. The model graph construction process for each set of changes (e.g., of the uninteresting/interesting/critical classes of changes) comprises the initialization of a corresponding graph with the first string representation of content, and the subsequent update of this initial graph with the graphs of the other content instances in the class. Specifically, given two graphs, G_1 and G_2, the first representing the training set of changes and the second a new instance, we create a single graph that represents the updated model graph G_1 with the graph of new evidence G_2. This creates a class model that acts as a representative graph for the class content instances. More than one model graphs should be created: one per feedback alternative of the user.

4.2 Ranking using User feedback

A methodology based on Support Vector Machine Regression (SVR) is assigning importance values to entity changes and learn the user model.

For the content C of changes, on the other hand, we first calculate the size-normalized value similarity between the n-gram graph G_C of a judged C, with respect to each of the n-gram graphs of the user model $G_U^{important}$, $G_U^{unimportant}$, $G_U^{critical}$. This similarity value, which lies between 0.0 and 1.0, indicates what part of the graph of C can be found in the corresponding graphs of the model of the user. This set of similarities S = $S6unimportant$, $S^{important}$, $S^{critical}$is the second constituent of the representation of a description D =¡ T,C ¿ of a change with respect to a user model. To use this set of similarities, we integrate them within the vectors of the type as new dimensions-features. Therefore, when n-gram graphs are used, the overall importance of a change is estimated based on the combined vector for type and content in an extended input vector space.

4.3 Advantages

- When the content methodology is used, most profiles (on which experiments have been performed) are feasible to learn.
- Very few iterations (¡ 10) are enough for the profile to be learned. Even the complex profile is learned in about 20 iterations.

4.4 Shortcomings

- In the case of the name-value-based attribute, performance is not very good probably because of the represention the name-value pair as a single string which generate noise in the n-gram graph pattern matching (because features of similarity are generated for them in common)

5 Text Classification

Text classification (TC), also known as text categorization, is the task of automatically detecting one or more predefined categories that are relevant to a specific document. Specifically, the n-gram graph approach has been developed to address the following challenges which exist in the traditional approaches.[18][19][20]

1. **Multilinguality:** Most representation models are language specific: to ensure high performance, they

   ne- tune their functionality to the language at hand. This is typically done with pre-processing techniques, such as lemmatization, stemming and word sense disam- biguation with the help of dictionaries (e.g., WordNet3). Social Media posts can be in any language, but they typically lack any metadata that denotes it.

2. **Sparsity** Social Media content solely comprises free-form text that is rather short in length, especially when compared to traditional Web documents, like Web pages. Due to size limitations, individual mes- sages typically consist of a few words, thus involving little extra information that can be used as evidence for identifying the corresponding category.

3. **Noise** Social Media posts are particularly noisy, due to their casual, real-time nature and their minimal curation. For example, users frequently participate in chats, posting their messages as quickly as possible, without verifying their grammatical or spelling correctness; incomprehensible messages can be simply corrected by a subsequent post.

4. **Evolving, non-standard vocabulary** A large part of the activity in Social Media pertains to informal communication between friends, who typically employ a casual ”communication protocol” (e.g., slang words and dialects) [21]. The limited size of their messages also urges them to shorten words into neologisms that bear little similarities to the original ones (e.g., ”gr8” instead of ”great”).

5.1 Application in Spam Filtering

The n-gram graph methodology has been used inspam filtering task using JInsect Application Suite. The task is the one described in CEAS 2008(http://www.ceas.cc/2008/challenge/). The task was to classify about 140000 e-mails as spam or ham, but in the process of classification feedback was given to the classifier, much like the case where a user provides feedback on what is considered to be spam. Given this scenario, a variety of approaches that used n-gram graphs as the means were used to represent the e-mails and their classes, to compare new e-mails to the classes and to update the class representations. In preliminary experiments, the most promising approach, which used a maximum of 20K characters from the e-mails, with no preprocessing at all, performed really well. The evaluation measures for the filter combined with the percentage of spam blocked and the filter's false positive rate. These experiments, together with a first place in the primary ranking concerning a stemmatology challenge (Computer Assisted Stemmatology Challenge), where the n-gram graphs were used as a clustering methodology to determine the derivation of different versions of a single text.

5.2 TC on Web documents of different categories

The TC technique involving n-gram graphs improves the performance of topic classification across all types of Web documents. This model goes beyond the established bag-of-words one, representing each document as a graph. Individual graphs can be combined into a class graph and graph similarities are then employed to position and classify documents into the vector space.

5.2.1 Advantages:

- **Accuracy** is increased due to the contextual information that is encapsulated in the edges of the n-gram graphs.

- **Efficiency**, on the other hand, is boosted by reducing the feature space to a limited set of dimensions that depend on the number of classes, rather than the size of the vocabulary.

- **Performance-** Studies [22] conducted over three large-scale, real-world data sets validates the higher performance of n-gram graphs in all three domains of Web documents.

- **Less sensitive to inherent document content type and better performance for demanding social media content** In experiments [22] comparing n-gram graphs with the established representation models over the three real-world data sets of our experimental study. The outcomes demonstrate the significantly higher performance of n-gram graphs, not just for Social Media content, but across all types of Web documents.

- **Language Neutrality/ No Preprocessing:** The performance of the term vector model can be significantly degraded by spelling mistakes , as well, which hinder the process of detecting and clustering together the different appearances of the same word. This leads to an extensively larger feature space (i.e., lower efficiency) as well as to lower effectiveness, due to noise. The character n-grams model improves on both disadvantages of the term vector one, constituting a language-neutral technique that is highly robust to noise (especially with respect to spelling mistakes).

- **Handling Sparsity:** The n-gram graph approach ameliorates the effect of sparsity by encapsulating contextual information in its edges, whereas the bag-of-tokens models make no provision for this challenge, relying exclusively on the inadequate features extracted from the sparse content.

5.2.2 Shortcomings

1. **Dimensionality:** A serious drawback, common to both vector space and n-gram models, is the curse of dimensionality: the number of features that they entail is usually very high - depending, of course, on the size of the corpus (i.e., number of documents). In absolute numbers, it is higher for the n-grams than for the term vector model (for the same corpus), increasing with the increase of n; the reason is that sub-word tokens are typically more frequent than whole words, and the larger the values of n is, the higher is the number of possible character combinations. This situation is particularly aggravated in the context of a highly diverse vocabulary: the more heterogeneous a document collection is - either with respect to the languages it comprises or the regional variations used by its authors - the higher is the number of features that these methods take into account.

2. **Time Complexity:** Another drawback is the time that is needed in order to construct a class-representative graph and to compute the graph similarities. The time complexity of these processes depends both on the size of n and the size of the input document collection which could be very high.

5.2.3 Classifying documents into sets based on Inherent characteristics

The performance of the traditional models of Text Classification (like bag-of-tokens model etc.) depends heavily on the inherent characteristics of the document collection at hand. In fact, their effectiveness is degraded by semantically incorrect (or incomprehensible) phrases and by spelling, syntactical and grammatical mistakes, as these characteristics introduce noise to the information conveyed by a document. However, not all types of documents convey the same levels of noise. Based on these characteristics, we can divide the web documents into three types:

1. **Curated documents :** which entail large documents of pure text (i.e., without notations) with standard, formal vocabulary and low levels of noise.

2. **Semi- Curated documents :** which are shorter in size, involve more noise, a slightly larger vocabulary, and plenty of hyperlinks.

3. **Raw documents :** which are rather telegraphic, noisy and rich in special notation.

The selected document types cause variation in the performance of text classification systems, not only in terms of effectiveness, but also of efficiency.

5.2.4 Techniques for using n-gram graphs in TC

To deal with them, we apply a novel, efficient and language-neutral representation method that is robust to noise: the n-gram graphs. It goes beyond the plain bag-of-tokens models by representing individual documents and entire categories as graphs: their nodes correspond to specific n-grams, with their weighted edges denoting how close the adjacent n-grams are found on average. In this way, it adds contextual information to the n-grams model, thus achieving higher accuracy. It also improves the time efficiency of learning, by addressing successfully the problem of the dimensionality curse". Documents are classified according to a limited set of graph similarity metrics, with the overall number of features depending on the number of classes, instead of the vocabulary size. A brief description of the technique is as follows:-

1. To represent a document d_i, we create a document graph G_{d_i} by running a window of size D_win over its textual content in order to analyze it into overlapping character n-grams. Any two n-grams that are found within the same window are connected with an edge $e^G_{d_i} \in E^G_{d_i}$ whose weight denotes their frequency of co-occurrence in the document. The document is, thus, transformed into a graph that - in addition to its n-grams - captures the contextual information of their co-occurrence.

2. This representation can also be employed for an entire topic (i.e., set of documents). In this case, however, the graph is derived from the merge of the individual document graphs, similarly to the concept of a centroid vector. The graph models of the topic's documents are merged into a single class graph through the update operator.

3. The similarity between documents and topics is estimated through the closeness of their graph representations. The following graph similarity metrics are used in this work:

   - **Containment Similarity (CS)** which expresses the proportion of edges of a graph G_i that are shared with a second graph G_j .
   - **Size Similarity (SS)** which denotes the ratio of sizes of two graphs
   - **Value Similarity (VS)** which indicates how many of the edges contained in graph G^i are contained in graph G^j , as well, considering also the weights of the matching edges.
   - **Normalized Value Similarity (NVS)** enhances VS by disregarding the relative size of the compared graphs.

4. To classify a document using the n-gram graphs model we first calculate the class graphs from the training instances. Each unlabeled document is then positioned into a vector space, as follows:

   - The document is represented as a graph (i.e., document graph.

- For every class, we compare the document graph with the corresponding class graph to derive the similarities that comprise the feature vector. In more detail, we extract 3 features from each comparison, one for each of the similarity measures CS, VS and NVS. The result, is that we get 3 similarity features per class, for our document.
- Given N class graphs, the resulting feature vector contains $3 \times N$ similarity-based features.

6 References

[1] R. Mihalcea. Graph-based ranking algorithms for sentence extraction, applied to text summarization. In Proceedings of the 42nd Annual Meeting of the Association for Computational Lingusitics (ACL 2004)(companion Volume). ACL, 2004.

[2] G. Erkan and D. R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. Journal of Artificial Intelligence Research, 22:457–479, 2004

[3] G. Erkan and D. R. Radev. Michigan at duc 2004 – using sentence prestige for document summarization. Proceedings of the Document Understanding Conferences Boston, MA, 2004.

[4] Jahna Otterbacher, Gune¸s Erkan, and Dragomir R. Radev. Using random walks for question-focused sentence retrieval. In HLT '05: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, pages 915–922, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

[5] Daniel Marcu. Theory and Practice of Discourse Parsing and Summarization, The. The MIT Press, 2000.

[6] S. Lamkhede. Multidocument summarization using concept chain graphs. Master's thesis, 2005

[7] Inderjeet Mani and Eric Bloedorn. Multi-document summarization by graph search and matching. In Proceedings of AAAI-97, pages 622–628. AAAI, 1997.

[8] Rene Witte, Ralf Krestel, and Sabine Bergler. Context-based multi-document summarization using fuzzy coreference cluster graphs. In Proceedings of Document Understanding Workshop (DUC), New York City, NY, USA, June 2006.

[9] E. Reiter and R. Dale. Building applied natural language generation systems. Natural Language Engineering, 3(01):57–87, 2000..

[10] E. Tsang. Foundations of constraint satisfaction. Academic Press San Diego, 1993.

[11] E.M. Voorhees. Query expansion using lexical-semantic relations. In Proceedings of the 17th annual international ACM SI- GIR conference on Research and development in information re- trieval, pages 61-69. Springer-Verlag New York, Inc. New York, NY, USA, 1994., Y. Qiu and H.P. Frei. Concept based query expansion. In Proceedings of the 16th annual international ACM

SIGIR con- ference on Research and development in information retrieval, pages 160169. ACM Press New York, NY, USA, 1993.

[12] A. Esuli and F. Sebastiani. Determining the semantic orientation of terms through gloss analysis. Proc. CIKM-2005, 2005.

[13] T. Pedersen, S. Banerjee, and S. Patwardhan. Maximizing semantic relatedness to perform word sense disambiguation. Su- percomputing institute research report umsi, 25, 2005.

[14] V. Rentoumi, V. Karkaletsis, G. Vouros, and A. Mozer. Senti- ment analysis exploring metaphorical and idiomatic senses: A word sense disambiguation approach. Proccedings of Interna- tional Workshop on Computational Aspects of Affectual and Emotional Interaction (CAFFEi 2008), 2008.

[15] L. Rabiner. A tutorial on hidden Markov models and selected applications inspeech recognition. Proceedings of the IEEE, 77(2):257-286, 1989.

[16] Y. Choi, E. Breck, and C. Cardie. Joint extraction of entities and relations for opinion recognition. In Proc. EMNLP, 2006 [17] G. Giannakopoulos, "Automatic summarization from multiple documents," Ph.D. dissertation, Department of Information and Communication Systems Engineering, University of the Aegean, Samos, Greece.

[18] F. Figueiredo, F. Belem, H. Pinto, J. M. Almeida, M. A. Goncalves, D. Fernandes, E. S. de Moura, and M. Cristo. Evidence of quality of textual features on the web 2.0. In CIKM, pages 909-918, 2009.

[19] S. Garcia Esparza, M. O'Mahony, and B. Smyth. Towards tagging and categorization for micro-blogs. In AICS, 2010.][S. Kinsella, M.Wang, J. G. Breslin, and C. Hayes. Improving

[20] Categorisation in social media using hyperlinks to structured data sources. In ESWC (2), pages 390-404, 2011.

[21] J. Eisenstein, B. O'Connor, N. A. Smith, and E. P. Xing. A latent variable model for geographic lexical variation. In EMNLP, pages 1277-1287, 2010.

[22] Giannakopoulos, George, et al. "Representation Models for Text Classification: a comparative analysis over three Web document types." Proceedings of the 2nd international conference on web intelligence, mining and semantics. ACM, 2012.

[23] A. A. Mohamed and S. Rajasekaran. Query-based summarization based on document graphs. 2006.