AN EVENT-BASED HIDDEN MARKOV MODEL APPROACH TO NEWS CLASSIFICATION AND SEQUENCING

A Thesis Submitted to
The Graduate School of Engineering and Sciences of
İzmir Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE

in Computer Engineering

by
Engin ÇAVUŞ

July 2014
İZMİR

We approve the thesis of **Engin ÇAVUŞ**

Examining Committee Members:

Assist. Prof Dr. Selma TEKİR
Department of Computer Engineering, Izmir Institute of Technology

Assist. Prof Dr. Tuğkan TUĞLULAR
Department of Computer Engineering, Izmir Institute of Technology

Assist. Prof Dr. Ayşegül ALAYBEYOĞLU YILMAZ
Department of Computer Engineering, Izmir Katip Çelebi University

14 July 2014

Assist. Prof Dr. Selma TEKİR
Supervisor, Department of Computer Engineering, Izmir Institute of Technology

Prof Dr. Halis PÜSKÜLCÜ
Head of Department of Computer Engineering

Prof Dr. R. Tuğrul SENGER
Dean of Graduate School of Engineering and Sciences

ACKNOWLEDGMENTS

I express sincere appreciation to my supervisor Assist. Prof Dr. Selma TEKİR for sharing her knowledge with me and guiding me throughout my thesis. Without her thesis proposal, support, encouragement, guidance and persistence this thesis would never have happened.

I should also express my appreciation to Kentkart company for giving me this opportunity, the managers understanding and encouragement to do my thesis.

Finally, I would like to thank to my family, my father Hasan ÇAVUŞ, my mother Fatma ÇAVUŞ, my sister Elvan and her husband Tarık and my dear nephew Sarper Efe GÜNDÜZ for their unlimited patience, support and love during the this thesis and all my life.

ABSTRACT

AN EVENT-BASED HIDDEN MARKOV MODEL APPROACH TO NEWS CLASSIFICATION AND SEQUENCING

Over the past years the number of published news articles have an excessive increase. In the past, there was less channel of communication. Moreover the articles were classified by the human operators. In the course of time the means of the communication increased and expanded rapidly. The need for an automated news classification tool is inevitable. The text classification is a statistical machine learning procedure that individual text items are placed into groups based on quantitative information.

In this study, an event based news classification and sequencing system is proposed, the model is explained. The decision making process is represented. A case study is prepared and analyzed.

ÖZET

OLAY TABANLI GİZLİ MARKOV MODELİ YAKLAŞIMI İLE HABER SINIFLANDIRMASI VE SIRALANMASI

Son yıllarda yayınlanan haber metni sayısında aşırı miktarda artış vardır. Geçmişte daha az iletişim kanalı bulunmaktaydı. Dahası, haber metinleri insanlar tarafından sınıflara ayrılıyordu. Zamanla iletişim araçları hızla arttı ve yaygınlaştı. Haber sınıflandırmasını otomatik olarak yapan bir yazılıma olan ihtiyaç kaçınılmazdı. Metin sınıflandırması istatistiksel bir makine öğrenimi işlemidir ki her bir metin elemanı nicel bilgilerine göre gruplara ayrılır.

Bu çalışmada olay tabanlı haber sınıflandırması ve sıralaması sistemi önerilmiştir ve ileri sürülen model anlatılmıştır. Karar verme işlemi anlatılmıştır. Örnek bir çalışma hazırlanmış ve incelenmiştir.

TABLE OF CONTENTS

LIST OF FIGURES

Figure **Page**

CHAPTER 1

INTRODUCTION

This thesis presents an application of Hidden Markov Models (HMM) to news classification problem. An HMM is a statistical tool for modelling generative sequences characterized by a set of observable sequences. The HMM structure is used for modeling stochastic processes where observable sequences of system have an underlying probabilistic dependence. So, the news articles are analyzed by an event based HMM approach for making probabilistic predictions about the news events.

The primary elements of the HMM are states and observations. The HMM's state parameter is types of events for an event based approach. The observation parameter is the news articles. The most known event based news categories are selected as event states. Two operations are applied to fill the event states sets with coherent and consistent keywords. For populating event sets the data are extracted from The New York Times. The most frequent words are treated as the keywords. Secondly, there are several web sites that quickly provide possible synonyms and also antonyms. These web sites get benefit from a reference work which is called Thesaurus. Thesaurus lists words according to the similarity of meaning. So, the state names are searched and related words are taken as a set. Moreover, both state sets are combined together to make a coherent and consistent data set.

At the beginning of the project, raw news data is planned to be used for tagging. However, some preprocessing need emerged. The Reuters wire agency published a news corpus for the researchers and the people working on the development of natural language processing. The idea of using this corpus is practical. The corpus has already been preprocessed and it has a standard structure. So the required elements for the HMM is acquired. In addition to the original corpus, Lewis et al. provides the term/document matrix and other information for the corpus.

As part of the preparation of a training set for the constructed HMM model, self-developed methods are used for matching the keywords of the event state sets and the elements of the news text. Subsequently, the keyword evaluation process is done more efficiently by Jaro Winkler tool. Moreover, a similarity measure is used to test

according to threshold. If the similarity Jaro Winkler reference is greater than threshold, then the words will tagged with the related class names.

As stated earlier, the proposed model is constructed around event types. Event types represent the model states and news articles the observations. There is no such event-based Hidden Markov Model adaptation in the news context. Although there exists the application of Hidden Markov Models in the area of news categorization [1] by news categories as states, the proposed technique is still original as the news categories and news events are at the different abstraction levels.

Firstly, a news text is given the system as an input. Then, the similarity of the news text elements and the generated news class items are measured. As a result of these processes, the training data is formed.

In the following parts, firstly the news analysis is going to be described and then the probabilistic approaches and the Markov Model is going to be introduced and finally the experimental work is going to be explained.

CHAPTER 2

NATURAL LANGUAGE PROCESSING

The Natural Language Processing (NLP) is a multidisciplinary field that computer science, artificial intelligence and the linguistics are concerned with the interactions between the human languages and computers. The process is basically deriving grammatical structure and meaning from natural language inputs by the help of computers. The rules of the target language and the task is important to perform suitable processes. The NLP [2] is beneficial when the subject is duplicate detection, database interface fields, supported instruction and tutoring systems.

There are some challenges in NLP applications development. The communication between the user and the computer may have some difficulties. The man already knows the natural language so he does not need to learn anything about an artificial language. Moreover, the user knows what he wants from the machine and he expresses himself in natural language. However, he can not know how to express himself to the machine.

The NLP is composed of four main parts : Morphology, Syntax, Semantics and Pragmatics.

2.1. Morphology

The natural language is composed of very large number of words and morphemes. The morphemes are known as the smallest grammatical units in a language. The morphology is the identification, analysis and description of the morphemes of a language. These morphemes could be root words, affixes, parts of the speech, intonations, and stresses. The morphology basically is a systematic description of words in a language. The identification of the grammatical context of a word is essential. For example, in English regular verbs have ground forms with a little modification from the origin, however the irregular verbs do not obey the modification rules and increase the language complexity.

2.2. Syntax

In natural language, the syntactical analysis refers to analysing a string of symbols according to the rules of the grammar. The user input is taken and then formal analysis is done to form parse trees. The syntactical relations of the words can be seen clearly in these trees.

2.3. Grammar

In most of the languages there are eight parts of speech (word classes). These are nouns, determiners, pronouns, verbs, adjectives, adverbs, prepositions and conjunctions. In English, the statements are composed of a noun phrase, a verb phrase and a preposition phrase. The subject of a sentence is identified by a noun which is represented by the noun phrase. The nouns are classified semantically as proper nouns (Turkey, China) and common nouns (frog, milk) or as concrete nouns(book, laptop) and abstract nouns(heat, prejudice). A verb phrase represents an action. The verb phrases may include an imbedded noun phrase along with the verb. The preposition phrase describes the noun or a verb in the sentence.

2.4. Parsing

The Parsing process determines if a sentence is valid for the current languages grammar rules. As it was mentioned in the Syntaxt part, parsing is the process of converting the sentence into a tree representation. The sentence tree starts with the sentence itself and then shows the verb phrases and the noun phrases. Afer that the articles, the adjectives and the nouns are labeled.

2.5. Semantics

Semantics is the study of the meaning of linguistic expressions [3]. In the Natural Language Processing environment the Semantics is basically determining the meaning of the sentence. The language can be a natural language such as English or

Turkish, or an artificial language like computer programming languages. Each language has some general rules that bring out the relationship between the words of the sentences and meaning. The structure of the sentence is determined in the parsing. So, this process gathers information for the informative analysis in order to determine which meaning was intended by the user. The Semantics establish a representation of the objects and the actions that a sentence is defining. Moreover, the detailed information including in a sentence such as adjectives, adverbs and the prepositions are provided.

2.6. Ambiguity

The meaining of an idea, statement or a claim may have more than one possible equivalence in human language. The types of ambiguity are lexical, semantic, syntatic and referential. The **lexical** ambiguity occurs when a word have more than one meaning such as “broad”. The word means “to get on” and “a flat wood” at the same time. The **syntactic** ambiguity means that the structure of the sentence is problematic. So, this situation refers to erroneous interpretations about the meaning of the sentence. If there is more than one possible meaning for a sentence, we can say that the **semantic** ambiguity exists there. The **referential** ambiguity exists when something is referred without explicitly naming in a sentence. Using words such as “it”,”he”,”they” sometimes makes the sentences impossible to resolve.

2.7. Applications of NLP

To reduce and identify the erroneous parts, the preprocessing [4] should be done. The applications are tokenization, stemming and stop word removal. **Tokenization** is the process of breaking a text into words and phareses. The steps of the process is removing punctuation, capitalization and any other modifications. The **Stemming** is the process of removing lexicall elements to reach the origin of the word. The input words are processed an simplified to the ground forms. There are some familiar stop words exists in the sentences and phareses such as “the”,”is”,”at”,”which”,”on”. So, the **Stop word Removal** is needed to be done to improve the searching time efficiency.

CHAPTER 3

PROBABILISTIC APPROACHES

3.1. Markov Model

The Markov Model is based on the Markov Property. In probability theory, the Markov property means the memoryless property of a stochastic process. Markovian property is satisfied if the conditional probability distribution of future states depends only on the present state. Moreover, this property means that the future states of the process does not depend on the past state history.

3.2. Markov Chain

A Markov chain has a set of states :

$$S = \{ s_1 , s_2 , \ldots , s_r \}$$

The S symbol as the starting state shows the initial probability distribution. The process starts with one of these states and moves from one state to another. So, each move is called a step. If the current state is s_i and then it moves to state s_j the probability is illustrated by p_{ij}. The p_{ij} values are called transition probabilities. The process can stay at the same state, so at this time the transition probability is p_{ii}. A transition probability matrix shows the probabilities of the particular transitions. Moreover, all possible states and transitions are included in the process. The process never terminates and also the system always has a next state. The system's current state is hard to predict because the states change randomly. However, the future steps of the system can be predicted by the statistical properties.

The first order Markov assumption means that the probability of an observation at time t depends only on the observation at time t-1. However, the second order Markov assumption states that the probability of observation at time t depends on the ones at times t-1 and t-2.

The collection of weather data [5] can be used to make weather forecast. The predictions about how is the weather like today are based on yesterday then the day before and so forth.

These probabilities are shown as

$$P(w_n \mid w_{n-1}, w_{n-2}, \ldots, w_1)$$

For example, having known today's weather, the tomorrow's weather is to be forecasted. So, this situation is a first-order Markov assumption [5] and in the Figure 1 the probabilities of the tomorrow's weather are shown.

		Tomorrow's Weather		
		Sunny	Rainy	Foggy
Today's Weather	Sunny	0.8	0.05	0.15
	Rainy	0.2	0.6	0.2
	Foggy	0.2	0.3	0.5

Figure 1. Probabilities of Tomorrow's weather based on Today's Weather

The table hereby monitors the assumptions about today's weather and the probabilities about the next day. If today is sunny, then the weather will be 0.8 probably to be sunny again, 0.05 probably to be rainy and 0.15 probably to be foggy.

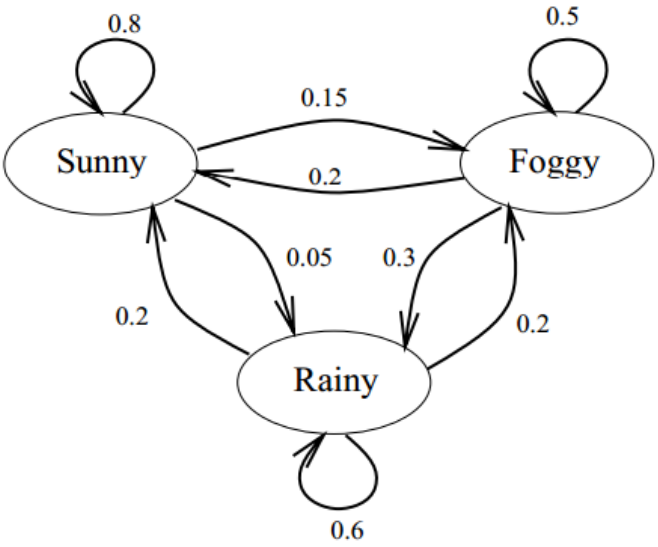

Figure 2. The automation of weather probabilities

The states here are Sunny, Foggy and Rainy. The transition probabilities are given in Figure 1 and the State Transition Diagram [5] is shown above as Figure 2.

3.3. The Urn Problem

The example is a generic one and called Urn and Ball model. There are three homogeneous urns and equivalent numbers of balls inside them. There is a genie in a room that is not visible to an observer. The room has U1, U2, U3,... number of urns and each urn contains number of colored balls B1, B2,B3,.... The proportion of each colored ball is different at each urn. So, the genie chooses an initial urn according to some random process. After that the genie draws a random ball from the urn and then the color of the ball is recorded as the observation. The ball is put back from where it has been taken. Then, according to the state transition probability, the new urn is chosen and a ball is selected again. This ball selection process generates a finite observation sequence of colors. The choice of the urn for the n-th ball depends only on the choice of the urn for the n-1 th ball.

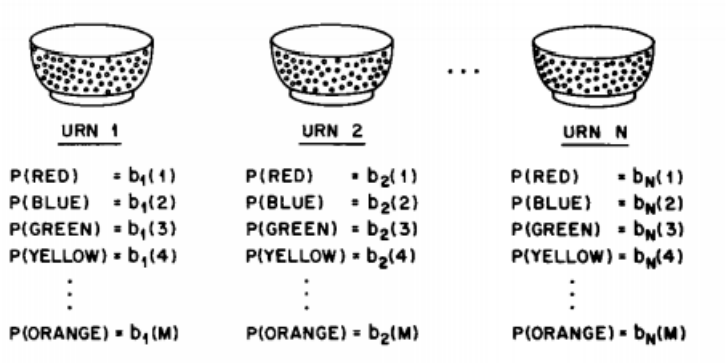

Figure 3. An N-State urn and ball model [6]

So the observer can see the sequence of the balls but not the sequence of the urns which the balls are drawn.

The typical observation sequence O = O1, O2, ..., Ot the probability of P(O | λ) is calculated based on the variables listed below on the Figure 4.

clock time	1 2 3 4 $\cdots T$
urn (hidden) state	$q_3 q_1 q_1 q_2 \cdots q_{N-2}$
color (observation)	$R\ B\ Y\ Y \cdots R$

T = length of the observation sequence (total number of clock times)
N = number of states (urns) in the model
M = number of observation symbols (colors)
$Q = \{q_1, q_2, \ldots, q_N\}$, states (urns)
$V = \{v_1, v_2, \ldots, v_M\}$ discrete set of possible symbol observations (colors)
$A = \{a_{ij}\}$, $a_{ij} = \Pr(q_j \text{ at } t+1 \mid q_i \text{ at } t)$, state transition probability distribution
$B = \{b_j(k)\}$, $b_j(k) = \Pr(v_k \text{ at } t \mid q_j \text{ at } t)$, observation symbol probability distribution in state j
$\pi = \{\pi_i\}$, $\pi_i = \Pr(q_i \text{ at } t = 1)$, initial state distribution

Figure 4. Formal model notation [7] for a discrete observation HMM

The HMM is represented as λ=(A,B,π). When the relative importances are analyzed, the π which represents the initial conditions, is the least important and B is the most important. B is directly related to the observation symbols with respect to the states.

3.4. Hidden Markov Model

In Markov Models [6], the states and the state transition probabilities are known and directly visible, on the other hand in Hidden Markov Models the states are not directly visible. The state transition probabilities are known. Since the transition probabilities are dependent on the states, HMMs allow the users to have the information of the possible next states.

The elements of Hidden Markov Model are:

1. In the model, there are finite number of states. Moreover, the states are not visible but the sequence of observations indicates the much likely states. For example, in the urn problem above the page, the states are the urns itself. The observations are the colored balls. The states are demonstrated by S = {S1,S2, ...,SN}.

2. The observation symbols can be changed based on the states. Each state is visualized with a separate alphabet character. The symbols are represented as V = {V1,V2, ..., VN}.
3. At each transition, the clock is altered and the state is changed based on its transition probability.

   The state transition probability distribution is denoted as $A = \{a_{ij}\}$ where

$$a_{ij} = P[q_{t+1} = S_j | q_t = S_i], \quad 1 \leq i, j \leq N.$$

4. The observation symbol probability distribution in state j is

   $B = \{b_j(k)\}$

   where

$$b_j(k) = P[v_k \text{ at } t | q_t = S_j], \quad 1 \leq j \leq N$$
$$1 \leq k \leq M.$$

5. The initial state distribution is represented by

   $\pi = \{\pi_i\}$

   where

$$\pi_i = P[q_1 = S_i], \quad 1 \leq i \leq N.$$

3.5. The Three Basic Problems for HMM

Since, there is a sequence of observations in the HMM model ($\lambda=(A,B,\pi)$), the probability of the observation sequence $P(O \mid \lambda)$ must be efficiently computed. This problem is called as The Evolution Problem of the HMM. The Forward Algorithm is the solution for the Evaluation problem [8].

The second case is choosing the most likely state sequence that produced the observations. This problem is known as The Decoding Problem of HMM. So, the solution of this problem is the Viterbi Algorithm.

The last problem is how to choose the model parameters (A,B,π) in order to maximize the probability of the observation sequences $P(O \mid \lambda)$. This is called as The

Learning Problem of the HMM. The Maximum Likelihood and Maximum Mutual Information Criterion Algorithms are the solutions for this problem.

CHAPTER 4

EXPERIMENTAL WORK

4.1. The Present Model

The proposed system is based on news classification and sequencing. The Hidden Markov Model approach is going to be utilized. The primary elements of HMMs are the states and the observations. An event based HMM uses the events as the states and the news data as the observations. For determining the major event related news categories the Wikipedia online encyclopedia is utilized. These news categories are used as event states. For the proposed model, the states are Ceremony, Happening, Party, Competition, Convention, Festival, Hazard and Crime. Moreover, the model parameters should be set. After the model set up, the training data set should be prepared. To built up training data, the event states must be populated with the coherent keywords. The meaning of the keyword and the context of the text affects the accurate tagging. The keywords are separated based on the elements of the sentence such as noun, verb and adjective to lessen the word-form ambiguities. The prepared training data are going to be used for the tagging. In the tagging phase, the observations are going to be matched with the states and each succesfully matched item takes the tag of the related state. The tags are given as the abbreviated state name and the letter of the respective part-of-speech word such as N for noun. The Competition, Ceremony, Convention and Crime keyword sets are composed of noun words. These tags are demonstrated as COMN, CERN, CONN and CRIN. The Festival event has the FESN and FESV for the noun words of the festival states and the verb formed words. The Hazard tag set is built both noun and verb keywords. So, the tags monitored as HAZN and HAZV. The PARN and PARV tags are used for the Party set to indicate the noun and verb formed words. The last tag set Happening has noun, verb and adjective words. These are illustrated as HAPN, HAPV and HAPA. See Appendix A for the event based keyword sets and related tags.

We determined event types as the states of the proposed Hidden Markov Model. In the news domain, the model is constructed. In probability theory, an event is

described as a set of outcomes of an experiment to which a probability is assigned [9]. So, the types of events are crucial to construct a consistent model. The opted states are Ceremony, Happening, Competition, Convention, Party, Hazard, Crime and Festival. The **Ceremony** is basically an event of ritual significance which is performed on a special happenings. The ceremony is an occasion that has a significance in the lifetime such as: birth, graduation, awarding, wedding, funeral. The **Happening** is especially an art based performance, event or a situation. So, music festivals and art festivals are good examples for the happenings. The **Competition** is a contest between individuals, groups and also animals and organisms. The Competition occurs when both sides exist in the same environment such as basketball. The both of the teams compete to score maximum point. The **Convention** can be considered as meeting. A group of people gathers at an arranged place and time to discuss some subject. Another state is **Party** that a host invites people to socialize, to have conversations between them. The **Hazard** is a natural or man-made tragedy that results in physical damage, loss of life or drustic change in environment. The **Crime** denotes an unlawful act punishable by a state [10]. A **Festival** is an event that is organized by a community to celebrate some common condition.

In the construction of the event-related word groups, we performed the following:

The elements of the event-related word groups should express the features of the individual group. The word sets should be populated with the coherent and relevant keywords. The USA's most popular news web site is The New York Times [11]. The web site is also very popular all around the world. By the records of "The Wall Street Journal" [12] web site receives more than 30 million unique visitors per month. For these reasons, The New York Times is chosen to fetch data for populating the event sets.

The New York Times frequently updates the news, entries and videos. The site also feeds lots of external web sites and blogs. To accomplish this the site has RSS section. The RSS stands for Rich Site Summary or Rally Simple Syndication. An RSS feed is a website's syndicated news feed to which the user can subscribe using an RSS or news reader.

The RSS feed (also known as an XML or news feed) is a listing of a website's content. It is updated whenever new content is published to the site. RSS feeds are

syndicated content. Syndication refers to the process that occurs when a publisher provides content in a form that can be consumed by software (like an RSS reader).

News readers "subscribe" to the feeds, automatically downloading lists of stories at a user-specified interval, and presenting them in the news reader. For example, when a user enters the New York Times RSS pages, one section appears instantly and offers “Subscribe Now”. If the user accepts the offer, then he will watch the latest news in the bookmark bar of the browser.

A news feed might contain a list of story headlines, a list of excerpts from the stories, or a list containing each story from the website. All news feeds have a link back to the website, so if a headline catches your eye, click on the link for that article to go directly to the related website.

The standard NY Times RSS structure contains title, link, description, language, copyright, publication date, last built date. If the news has any image then the rss will has image tag and the title of the image, the URL of the image and the link of the image. The xml structure can be demonstrated as:

<title> Joanna Brooks, Marcus Klostermeyer: United as They Journeyed Back in Time </title>
<link>
http://rss.nytimes.com/c/34625/f/642564/s/3b46f8ba/sc/8/l/0L0Snytimes0N0C20A140C0A60C0A80Cfashion0Cweddings0Cunited0Eas0Ethey0Ejourneyed0Eback0Ein0Etime0Bhtml0Dpartner0Frss0Gemc0Frss/story01.htm
</link>
<description> The bridegroom grew interested as the bride researched her family’s history through the Holocaust </description>
<language> en-US </language>
<copyright>Copyright 2014 The New York Times Company</copyright>
<pubDate>Sun, 08 Jun 2014 01:47:30 GMT</pubDate>
<lastBuildDate>Sun, 08 Jun 2014 01:47:30 GMT</lastBuildDate>
<ttl>2</ttl>
<image>
<title>NYT > Art & Design</title>
<url>http://graphics8.nytimes.com/images/2014/06/08/fashion/weddings/08BROOKS/08BROOKS-moth.jpg</url>
<link>http://www.nytimes.com/2014/06/08/fashion/weddings/melissa-weiner-and-ildefonso-de-jesus-jr.html?partner=rss&emc=rss</link></image>

The event based data sets can be populated by the help of RSS tags data. Moreover, the RSS feed links which are compatible with the model are found. To collect data from the tags, we need to have a website that can do this job for us. So, a script is prepared to fetch data from specific tags such as title and description. The description tag stores precious and brief data. Instead of having a long news text, we can understand the core details from the description data. For each event state, the title and description tags of each individual news article is stored for thirty days. We have a collection of data however the system needs to have the keywords which indicate the event types mostly. For this reason, another software is prepared to determine the most frequent words. The frequency in a document basically means the number of times the term occurs in the document. Furthermore, the most frequent words are determined and assigned to the related event sets.

In the calculation of the most frequent words in text, the tf-idf scheme [13] is utilized. The tf-idf refers to the *term frequency-inverse document frequency*. The tf-idf weight is a statistical measure that shows how important a word is in a document collection or corpus. In information retrieval and text mining processes, the tf-idf weight is used. The importance increases proportionally to the number of times a word appears in the document. The tf-idf weighting is used by the search engines in scoring and ranking a document's relevance to the word which is searched.

Generally the tf-idf weighting process is composed of two steps. The first one is calculating the Term Frequency. The second process is evaluating the Inverse Document Frequency. The Term Frequency refers to the number of times a word appears in a document, divided by the total number of words in that document. Since the documents have different lengths, a term may appear much more times in longer documents.

The TF formulation for term t:

TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document)

The Inverse Document Frequency (IDF) evaluates the importance of a term in the corpus.

The IDF formulation for the term t :

IDF(t) = log_e (Total number of documents / Number of documents with term t in it)

The TF-IDF weighting is the product of the term frequency and the inverse document frequency.

The TF-IDF formulation for the term t and document d :

TF-IDF (t,d) = TF (t,d) x IDF (t)

Example TF-IDF :

Consider a document containing 100 words and the “cost” word appears 8 times.
The term frequency (TF) for “cost” is then (8 / 100) = 0.08.
Assume that the system have 10 million documents and the word “cost” appears in one thousand of these.
The inverse document frequency (IDF) is calculated as
log(10,000,000 / 1,000) = 4
As a result of this, the tf-idf weight is the product of TF and IDF: 0.08 * 4 = 0.32.

In order to enrich the collection of words for each event type, used synonyms, antonyms, and dictionary definitions etc. There are several web sites that quickly provide possible synonyms and also antonyms, dictionary definitions and usage examples based on the users search. The user enters a word or a phrase and then searches. The answer of the query results in the synonyms and the antonyms of the input. Moreover, the resultant words are shown with their related words. These web sites get benefit from a reference work which is called Thesaurus. Thesaurus lists words grouped together according to similarity of meaning. The aim of the work is to help the user to find the most suitable words or phrases.

Although including synonyms, a thesaurus should not be taken as a complete list of all the synonyms for a particular word. The entries are also designed for drawing distinctions between similar words and assisting in choosing exactly the right word.

4.2. Tagger Construction For The Training Set

The news data could have some abbreviations and daily language words that the dictionaries may not cover. So, the user extracted keyword sets deal with these kinds of data. For example, people may encounter “celeb” word in newspapers and news web sites however the dictionaries do not have such a word. The online suggested words are in the grammatical form. The first process gives us consistent keywords for each event type but some of them are ungrammatical.

When both of the keyword sets are checked, they have some identical words. So, to set up a comprehensive keyword set, the user extracted keywords and the suggested word groups are combined. The combined keyword set covers both regular words, abbreviations and daily language words. It is expected that using this kind of well-grounded keyword set will increase the success of matching process.

Until this section we dealt with the event states and the keyword selection processes. To accomplish the news classification, we both need to have event based keyword sets and also news data sets. The Reuters is one of the international wire agency that has a news collection. The news data are in the xml format. The tags are the news id, date, title, headline, the text of the news and the copyright. An example of a news xml is shown below.

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<newsitem itemid="476023" id="root" date="1997-03-31" xml:lang="en">
<title>ESTONIA: ESTONIA-TALLINN STOCK EXCHANGE TAKES NEW MEMBERS.</title>
<headline>ESTONIA-TALLINN STOCK EXCHANGE TAKES NEW MEMBERS.</headline>
<dateline>TALLINN 1997-03-31</dateline>
<text>
<p>The Tallinn stock exchange board accepted two more members during their board meeting last week, the Tallinn Stock exchange said.</p>
<p>Hansa Investments, 50 percent owned by Hansapank, will become a member of the stock exchange, Helo Meigas, Managing Director of the Tallinn Stock exchange, told Reuters.</p>
<p>The other company, Vaarpaberiparsnike, was also approved as a member, Meigas added.</p>
<p>-- Tallinn newsroom +372 630 8400</p>
</text>
<copyright>(c) Reuters Limited 1997</copyright>
```

The news data set can be filled with the text section of the Reuters xml. So, it is needed to extract the text elements from each news xml. An appropriate script is written and the data are extracted from the required fields. Hereby, we have a collection of news data that builts a corpus. Moreover, the texts are raw texts what means they have stop words, punctuations and derived words. The Indexers and Analyzers can be used to get rid of these issues. The Lucene [14] tool can be used for full text searching. Lucene is composed of a set of libraries and file formats. Indexing basically means that each word's frequency and the addresses (as a pointer) are stored in a data structure. By the help of Lucene, data are transformed into an index data structure.

In Lucene-specific solution, to create an index, the first thing to do is to create an IndexWriter object. The IndexWriter object is used to create the index and to add new index entries to this index. You can create an IndexWriter as follows :

IndexWriter indexWriter = new IndexWriter("index-directory", new StandardAnalyzer(), true);

The first parameter specifies the directory in which the Lucene index will be created, which is index-directory in this case. The second parameter specifies the "document parser" or "document analyzer" that will be used when Lucene indexes your data. The third parameter tells Lucene to create a new index if an index has not been created in the directory yet.

The job of the Analyzer is to "parse" each field of your data into indexable "tokens" or keywords. There are several types of analyzers. The most known analyzers are Standard Analyzer, White Space Analyzer, Stop Analyzer and Snowball Analyzer. The StandardAnalyzer used for filtering Standard Token Filter, Lower Case Token Filter and Stop Token Filter. The Lower case token filter normalizes the token text to lower case. The Stop token filter removes the stop words from token streams. The Stopwords token filter has some settings such as, stopwords, stopwords_path, ignore_case and remove_trailing. The stopwords tag refers to the list of the stop words that can be determined by the user. The stopwords_path tag refers to the configuration file location address. The ignore_case refers to a boolean condition. If the value is true then all the words are set to lower case. The remove_trailing is set to false in order not to ignore the last term of a search if it is a stop word. This is very useful for the completion suggester as a query like green a can be extended to green apple even though you remove stop words in general.

The text searches can be done in Lucene Indexers. The SearchFiles class is used for the searching. This search class works with the IndexSearcher, StandardAnalyzer and the QueryParser.

The query parser is constructed with an analyzer to interpret users query text. The word boundaries are found and the useless words are eliminated such as “a, an, the”. The searcher takes the query parsers, output data and an integer to show the maximum number of the hits.

The Reuters released a news corpus for the researchers and the people working on the development of natural language processing, information retrieval or machine learning. The corpus is composed of the Reuters News articles.

The provided corpus is known as RCV1-v2/LYRL2004 [15] which has text categorization test collection that is composed of online appendices. The RCV1-v2/LYRL2004 test collection is made up of a large number of files, which takes the form of 18 On-Line Appendices to the LYRL2004 article.

On-Line Appendix 12 consists of ten ASCII files that contain tokenized documents. The five of the files contain the exact RCV1-v2 token files. These files are used for training and testing phases for the supervised learners in LYRL2004. Four of the files contain tokenized test sets and the fifth one contains the training set tokenized documents.

The number of documents in each file is:

lyrl2004_tokens_test_pt0.dat : 199328 test documents

lyrl2004_tokens_test_pt1.dat : 199339 test documents

lyrl2004_tokens_test_pt2.dat : 199576 test documents

lyrl2004_tokens_test_pt3.dat : 183022 test documents

lyrl2004_tokens_train.dat : 23149 training documents

There are 23,149 training documents and 781,265 test documents in these files, for a total of 804,414 documents, i.e. all the documents from RCV1-v2 as defined in LYRL2004. The documents have been tokenized, stopworded, and stemmed. Most but not all punctuation was removed during stemming.

A document has the format:

.I <did>

.W

<textline>+

<blankline>

where we have:

<did> : Reuters-assigned document id.

<textline> : A line of white-space separated strings, one for each token produced by preprocessing for the specified document. These lines never begin with a period followed by an upper case alphabetic character.

<blankline> : A single end of line character.

Each line that begins with ".I" indicates the start of a new document.

Here's an example of the tokenized document file format:

.I 1
.W
now is the time for all good documents
to come to the aid of the ir community

.I 2
.W
i am the best document since i have only one line

.I 3
.W
no i am the best document

4.3. Performing the Tagging Operation

The news data files which are in a raw form need to be indexed and analyzed. The Lucene is a full text searching engine which is developed by Apache. Moreover, there are built in analyzers for filtering the white spaces, stop words and punctuations. Firstly the raw files directory and the processed files directory are declared. After that the raw file is read, analyzed and indexed.

Although, the indexed files can be generated by the help of Lucene [14], the Reuters wire agencies news corpus can be used. The corpus which is known as RCV1-v2/LYRL2004 is utilized as the news resource. As soon as the system utilizes the Reuters news corpus, we do not need to use the raw news articles and Lucene mechanism. Furthermore, all the preprocess parts are passed. The news data do not need to be indexed because it is already indexed. Also, the analyzers are impractical because the corpus data are already analyzed. Moreover, the stopwords and punctuation removal do not required since the system uses Reuters news corpus. To use news data, a string array is declared and filled with a part of the corpus data.

Each individual event state is declared as string array. So, the related keywords are filled according to the suitable event states.

To accomplish the matching, each item from event set is compared with the news data items. The string comparator function equalsIgnoreCase is used for comparing two strings ignoring the case considerations. Two strings are considered equal if they are of the same length and corresponding characters in the two strings are equal. The method returns true if the argument is not null and the Strings are equal, ignoring case; false otherwise.

The syntax of this method:

public boolean equalsIgnoreCase(String anotherString)

anotherString -- the String to compare this String against

After the matching phase, it is observed that some of the words are missed and the success rates could be better. For example the missed words are the stem form of the keywords that was in the state set. Another problem is the daily language usage. The words can be in the shorten form. For example celebrity word is used as celeb. So, if the state sets does not cover this kind of words the matcher will fail. To overcome this problem, each keyword from state set is splitted into the sub strings. Each word taken as

the first three letter of itself and then the matcher function tries to match the splitted keywords with the news items in a loop. After each matcher call, the letter length is increased until the keywords length is reached. Using the splitted forms of the keywords increase the matchers performance. Moreover, if a word from a known set is matched with the news element, the word is tagged as " word/set tag". The ones which are not matched tagged as "word/O". The "O" tag refers to the "Other" word. The set tags are CER, HAP, HAZ, COM, CON, CRI, FES and PAR.

Example : collect/O bowl/O pass/HAP over/HAP shot/O craig/O put/O flow/O fourth/O sing/O gave/O outfield/PAR push/O moham/O field/COM stump/O early/O start/CER

The similarity between two strings can be measured by automated tools. The Jaro Winkler is one of the best distance algorithm which is applicable on strings. The higher the Jaro Winkler distances for two strings indicate the more similar strings. The Jaro Winkler similarity score is between 0 and 1. So, the Jaro Winkler distance is used to evaluate the similarity between the keywords and the news elements. As it is mentioned at The Distance Algorithms part, the distance function takes two strings. One of the input string is given from the state sets and the second string is taken from the news data. So, the first item of the event state is compared with each sequential item from news data and the distance is calculated for each comparison. Moreover, a new threshold value is declared and set to 0.89. To do more accurate tagging, the words are taken as keyword whose comparison value is greater than the threshold value.

Since the Jaro Winkler distance algorithm is used, the keywords do not need to be splitted as it was in the second version. Item by item comparison is not made by the user, Jaro handles it.

To gain a generic structure, the keywords and news data are read from seperate directories. The system may need to have some alterations. The event states may need some additional sets or the sets need more keywords. The proposed solution should operate when this kind of changes are required.

The tokenized news article document known as RCV1-v2/LYRL2004 has lots of news text. The news are seperated by their id numbers in the Reuters news corpus which is in the dat file format. Each news article is needed to be compared with the event sets. So, the news corpus is splitted into individual documents. For the training phase, the first part of the corpus is utilized. That part is known as lyrl2004-non-v2_tokens_test_pt0. This part has 671 documents however the first 100 documents are taken to built the training document. The splitted news files are stored in a directory.

The stored news files directory is declared in the system. The news files can be changed in any time or the system can use more or less data with respect to the subject. Each individual event state keywords which are read from directory has already assigned to sets. So, we have eight individual event set now. In a for loop, each seperate news data is read and assigned to set. The news set items are respectively compared with the event set items. The Jaro Winkler distance is evaluated for each item pair. If the result score is higher than the threshold value, the news item is added to a new set. The new set is called SET_PROX keeps the news words that have the maximum proximity with the event set elements. So, SET_PROX is going to be used for the tagging. The SET_PROX elements are tagged with the set tags as it is told in the second version. The "word/set tag" tagging structure is kept the same. There is also another set initialized to store the news elements which have grater score value from the threshold. This set is called SET_OTHERS which is going to be used for tagging the words that has lessen value from the threshold. From each set, all the elements above the threshold value are added to SET_OTHERS set. To tag the elements belov the threshold value as "Others", the collection of high proximity words are extracted from the news set. The resultant set words labeled with " /O" tag. So, the structure for tagging the others items is " word / O ".

As a result, collection of tagged elements from seperate event states are combined and added a set. Moreover, the irrelevant words are also tagged and added to the same set. The resultant set may have tags from seperate event states and also "/O" tags for the irrelevant words.

Pseudo Code For Matching Algorithm

```
Form SET_NEWS set
Form SET_PROX set
Form SET_TOKEN set
Form SET_OTHERS set
Form SET_NEWS_GENERAL set
Form SET_PROX_REMOVE set
Form SET_NEWS_OTHERs set
Form Event Sets for each event state
FOR  i = 0,.......,K       Let K is the number of elements in Ceremony event set
        FOR  j=0,....., T     Let T is the number of elements in news set
```

```
                Compute Jaro Distance AS Score
                IF Score > Threshold
                        Add news keyword to SET_PROX _CER
        END
END
FOR  each item in SET_PROX_CER
        Add "item + / Set" tag to SET_TOKENS
        Add items to SET_PROX_REMOVE
END
FOR  each item in SET_TOKENS
        Add items to SET_PROX
END
RemoveAll SET_PROX_REMOVE items from SET_NEWS_GENERAL  AS
SET_NEWS_GENERAL
FOR each item in SET_NEWS_GENERAL
Add "item + /O"  tag to SET_NEWS_OTHERS
END
Print SET_NEWS_OTHERS
Print SET_PROX
```

4.4 Testing the Model on the Reuters Corpus

The tagged Reuters news articles are achieved by the result of tagging process. The suitable keywords are opted for each matched news items. So, the Hidden Markov Model is needed be built and an automated tagging mechanism should be established. To accomplish this, the Java Text Mining Toolkit (JTMT) [16] is utilized. In JTMT project, the supervised learning technique [18] is used. So, the tagged training documents are going to be used to test the articles. The tags of the training document items are known and these tags are going to be used to classify the unlabeled articles.

The JTMT has built in Hidden Markov Model tagger classes based on Part Of Speech(POS) [19]. The states have the names of the POS tags such as nouns, verbs, adjectives, adverbs, etc. The experimental work must be compatible with the existing

HMM POS tagger class. Moreover, the POS keywords are altered with the event state category names.

In JTMT project training documents are stored in a directory. Initially these documents are suitable for POS tagging. Afterwards, the tagged news articles are replaced with the POS tagging articles. So, the built in HMM POS tagger class is converted into an event based HMM tagger class. As a result, the model uses the tagged news articles as training articles and the event related keywords as tags.

```
State 0
  Pi: 0.02631578947368421
  Aij: 0,02 0,03 0,04 0,01 0,01 0 0,015 0,005 0,87
  Opdf: Integer distribution --- 0 0 0 0 0 0 0 0 0,028 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

State 1
  Pi: 0.03508771929824561
  Aij: 0,022 0,051 0,051 0,022 0,027 0,003 0,03 0 0,794
  Opdf: Integer distribution --- 0 0 0 0 0 0 0 0 0,016 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

State 2
  Pi: 0.03508771929824561
  Aij: 0,028 0,032 0,014 0,011 0,007 0,004 0,018 0,004 0,883
  Opdf: Integer distribution --- 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0,004 0 0 0 0 0

State 3
  Pi: 0.0
  Aij: 0,039 0,029 0,029 0 0,126 0 0,039 0,039 0,699
  Opdf: Integer distribution --- 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0,111 0 0 0 0 0 0 0 0

State 4
  Pi: 0.07017543859649122
  Aij: 0,018 0,048 0,012 0,018 0,018 0,006 0 0,012 0,867
  Opdf: Integer distribution --- 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0,072 0 0 0 0 0 0 0 0

State 5
  Pi: 0.008771929824561403
  Aij: 0,024 0,083 0,012 0 0,024 0,036 0,036 0 0,786
  Opdf: Integer distribution --- 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

State 6
  Pi: 0.043859649122807015
  Aij: 0,019 0,065 0,016 0,022 0,034 0,006 0,05 0,006 0,782
  Opdf: Integer distribution --- 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0,003

State 7
  Pi: 0.0
  Aij: 0,061 0,082 0,02 0 0,061 0 0,082 0 0,694
  Opdf: Integer distribution --- 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

State 8
  Pi: 0.7807017543859649
  Aij: 0,021 0,035 0,028 0,01 0,014 0,009 0,033 0,005 0,844
  Opdf: Integer distribution --- 0 0 0 0,001 0,001 0 0 0 0 0 0 0 0 0 0 0,001 0 0,002 0
```

Figure 5. The Hidden Markov Model parameters representation.

4.4.1 Calculating the Model Parameters

Since, the training documents and the suitable keywords are determined, we can built the event based Hidden Markov Model. As it was mentioned before the HMM parameters are, Q, A, B and π. The states of the model represented with Q. The state transition probabilities is demonstrated with A. Furthermore, B refers to observation probabilities.

In Figure 5 “State 0” refers to Ceremony, “State 1” refers to Happening, “State 2” refers to Hazard, “State 3” refers to Convention, “State 4” refers to Competition, “State 5” refers to Crime, “State 6” refers to Party, “State 7” refers to Festival and “State 8” refers to Others. The Pi value for each state refers to initial probabilities (start probability) respectively. “Aij” denotes the state transition probabilities and the “Opdf” denotes the observation probabilities.

4.4.2 Optimizing the Model Parameters

In the event based Hidden Markov Model, the supervised learning techniques are used. For this reason, the training documents should be prepared more carefully. Since the system is built on event domain, the articles for the training must be selected from the same domain. Moreover, the training documents composed of the articles that contain suitable tags. The event based keywords should be selected more accurately. The coherent keyword selection affects the training data quality and the training data affects the whole model. Another point is coherent articles selection.

Both the keyword and training data selections affect the Hidden Markov Model parameters truthfulness. When an occupant examine the HMM parameters (Figure 5), he will make more correct interpretation about the system.

CHAPTER 5

CONCLUSION

The main goal of this thesis is to handle the need for an automated news classification tool. Moreover, to handle this issue an event based Hidden Markov Model is developed to classify the news articles. The news articles are analyzed by an event based HMM approach for making probabilistic predictions about the news events. The experimented news categories are Ceremony, Happening, Party, Competition, Convention, Festival, Hazard and Crime. The news categories are used as states of the HMM. To improve the classification accuracy, several preprocessing techniques are applied to news articles. However, the Reuters wire agencies text categorization research files are utilized. The use of the Reuters tokenized documents is reduced the training computational time. The Jaro Winkler Distance metric is used for monitoring the similarity between the elements of the news articles and the elements of the event based keyword set. The default Jaro Winkler Reference is set to 89% to acquire more relevant results. Since we have used a small dataset of news article sequences, the accuracy can be enhanced by increasing the size of the dataset. Furthermore, the amount of the keywords in the event sets can be increased to enhance the accuracy. As a result of experimental works, an event based Hidden Markov Model is constructed. The proposed systems model parameters are calculated and illustrated at Figure 5.

REFERENCES

[1] Krishnalal G, S Babu Rengarajan, K G Srinivasagan, A New Text Minig Approach Based on HMM-SVM for Web News Classification International Journal of Computer Applications (0975 – 8887), 2010.

[2] Saad Ahmat , Tutorial on Natural Language Processing. University of Northern Iowa, Artificial Intelligence (810:161) Fall 2007.

[3] Richard H. Thomason, What is Semantics[Internet], http://web.eecs.umich.edu/~rthomaso/documents/general/what-is-semantics.html (accessed date : 2 May 2014).

[4] Mita K. Dalal, Mukesh A. Zaveri, Automatic Text Classification: A Technical Review. International Journal of Computer Applications (0975 - 8887) Volume 28 – No. 2, August 2011.

[5] Eric Fosler-Lussier, Markov Models and Hidden Markov Models A Brief Tutorial. TR-98-041. International Computer Science Institute, December 1998.

[6] Lawrence R.Rabiner, A Tutorial on Hidden Markov Models and Speech Applications in Speech Recognition. IEEE VOL.77 NO.2, February 1989.

[7] L. R. Rabiner, B. H. Juang, An introduction to Hidden Markov Models. IEEE ASSP MAGAZINE JANUARY 1986.

[8] Prakash M Nadkarni, Lucila Ohno-Machado, Wendy W Chapman, Natural language Processing: An Introduction. International Computer Science Institute, Yale University School of Medicine. July 2011.

[9] Leon-Garcia, Alberto, Probability, Statistics and Random Processes for Electrical Engineering. Upper Saddle River, NJ: Pearson,2008.

[10] Oxford English Dictionary Second Edition on *CD-ROM*. Oxford: Oxford University Press. 2009

[11] The New York Times[Internet], http://www.nytimes.com/services/xml/rss/index.html,(accessed date : 1 September 2013).

[12] Adams, Russell, "New York Times Prepares Plan to Charge for Online Reading". *The Wall Street Journal*. Retrieved January 26, 2011.

[13] Juan Ramos , Using TF-IDF to Determine Word Relevance in Document Queries Department of Computer Science, Rutgers University, 23515 BPO Way, Piscataway, NJ, 08855.

[14] Apache Lucene Core[Internet], http://lucene.apache.org/core/, (accessed date : 2 May 2014).

[15] Lewis, D. D.; Yang, Y.; Rose, T.; and Li, F. RCV1, A New Benchmark Collection for Text Categorization Research. Journal of Machine Learning Research,5:361-397,2004.

[16] Java Text Mining Toolkit[Internet], http://jtmt.sourceforge.net/, (accessed date : 10 September 2013).

[17] Hang-Bong Kang, Affective Content Detection using HMMs, Dept. of Computer Engineering The Catholic University of Korea.

[18] Cord**,** Matthieu**,** Cunningham**,** Pádraig, Machine Learning Techniques for Multimedia. Case Studies on Organization and Retrieval. Series: Cognitive Technologies, 2008

[19] Ronan Collobert, Jason Weston, L´eon Bottou, Michael Karlen , Koray Kavukcuoglu Pavel Kuksa, Natural Language Processing (Almost) from Scratch, Journal of Machine Learning Research 2493-2537 December 2011.

APPENDIX A

FORMAT OF KEYWORD SETS

FESN FESTIVAL festival noun

carnival, elebration, circus, fair, feast, festivity, fete, fiesta, holiday, jamboree, revelry, wassail, saturnalia, rodeo, carousal, madri gras, festiveness

FESV FESTIVAL festival verb

Fete

COMN COMPETITION competition noun

antagonism, bout, capitalism, competitor, conflict, festival, field, game, meet, meeting, opposition, race, rival, scramble, side, strife, warfare, rodeo, candidacy, one-upmanship, contender, corrival, emulation, polarity

CERN CEREMONY ceremony noun

anniversary, celebration, commemoration, courtesy, cult, custom, event, exorcism, formality, jamboree, manners, memorial, pageantry, parade, performance, pomp, punctilio, requiem, sacrament, service, splendor, state, toast, wassail, red carpet, solemnity, starch, obsequies, set piece, gentelness, good form, mannerliness, politeness

HAZN HAZARD hazard noun

accident, adventure, bet, chance, difficulty, disaster, enterprise, exposure, fortune, hardship, impediment, jeopardy, lot, luck, mishap, peril, possibility, probability, romance, speculation, threat, undertaking, insecurity, endangement, happanchance, imperilment

HAZV HAZARD hazard verb

bet, chance, compromise, dare, endanger, expose, gamble, guess, imperil, jeopardize, lay, menace, pawn, play, risk, speculate, stake, undertake, venture, wager, hypothecate

HAPN HAPPENING happening noun

adventure, affair, arrival, business, chance, circumstance, coincidence, condition, contingency, enterprise, episode, eventually, fact, incident, occasion, operation, opportunity, phenomenon, possibility, proceedin, gragething, transaction, undertaking, hap, happenchance, doings, program, current affairs, hard news, line up

HAPV HAPPENING happening verb

arise, befall, begin, come, come off, develop, evantuate, exist, fall out, fare, go, go off, go on, materialize, overtake, pan out, pass, rise, betide, work out, hap

HAPA HAPPENING happening adjective

modish, up-to-date, begun, cooking, in vogue

CONN CONVENTION convention noun

bargain, behavior, bond, caucus, code, compact, company, concord, congress, contract, convocation, council, covenant, culture, custom, decorum, doctrine, ethics, etiquette, fashion, form, forum, gathering, habit, heritage, institute, jamboree, meeting, mode, pact, practice, principle, punctilio, rally, ritual, seminar, stereotype, symposium, tradition, transaction, treaty, usage, politeness, junta, unwritten law, rightfulness, rightness, protocol

PARN PARTY party noun

band, barbecue, bee, bevy, blast, body, brigade, bunch, celebration, circle, cluster, coalition, combination, company, concentration, corps, creature, crew, crowd, crush, cult, detachment, detail, dissipation, entertainment, expedition, faction, festivity, fling, function, gala, gang, gathering, group, individual, jamboree, machine, mortal, movement, organization, orgy, outfit, part, participant, partnership, person, persuasion, revel, revelry, ring, school, sect, side, spree, team, treat, troop, troupe, wassail, saturnalia, mixer, shindy, get-together, bacchanal, festiveness, ruck, visitant

PARV PARTY party verb

celebrate, have a ball, regale, wassail

CRIN CRIME crime noun

abomination, abuse, atrocity, carnage, con, contraband, corruption, delinquency, depredation, enormity, evil, fault, foul, play, guilty, heist, holdup homicide, infraction, iniquity, injustice, lapse, larceny, manslaughter, murder, offense, pity, racket, revolution, sacrilege, scandal, sin, stickup, theft, transgresson, treason, trespass, wrong, doingwrong, irregularity, lawbreaking, evildoing, impingementm, isapplication, misappropriation, mishandling, mistreatment, misusage, montrousness, obstrusion, peccancy, pilferage, profanation, seditiousness, traitorousness, unfairness, unjustness, villainousness