

**“NORMALISING MACRONIC TEXT INTO A UNIFORM
LANGUAGE”**

A PROJECT

*Submitted in complete fulfillment of the requirements for the award of the
degree of*

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE ENGINEERING

Under the supervision of

Dr.Rajni Mohana

By

Nimisha Nadda(121112)

to



JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY

WAKNAGHAT SOLAN – 173 234

HIMACHAL PRADESH INDIA

May, 2016

CERTIFICATE

Candidate's Declaration

I hereby declare that the work presented in this report entitled “**Normalizing macaronic text into a uniform language**” in complete fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering** submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from August, 2015 to May, 2016 under the supervision of **Dr. Rajni Mohana** Assistant Professor in the department of Computer Science & Engineering.

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Nimisha Nadda, 121112

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Dr. Rajni Mohana

Assistant Professor

Computer Science & Engineering

Dated: 30th May, 2016

ACKNOWLEDGEMENT :-

I express my sincere thanks to Dr. Rajni Mohana, Assistant Professor in the Computer Science & Engineering department for her support and guidance for doing the project. It was her support and continuous guidance that helped me to achieve what I have achieved in this project.

I would also like to thank Sukhnandan Ma'am, for helping me out understanding the software and hence, the project better. It was her guidance that helped me achieve what I have.

CONTENTS

CHAPTER 1. INTRODUCTION.....	1
1.1 Introduction	1
1.1.1. Natural Language Processing.....	3
1.1.2 Software Information: RapidMiner.....	9
1.2 Problem Statement	10
1.3 Objectives	10
1.4 Methodology	11
1.4.1 Identifying and Collecting Dataset.....	11
1.4.2 Filtering.....	11
1.4.3 Tokenization.....	12
1.4.4 Standardization	12
1.4.5 Generating bi-grams.....	12
1.4.6 Retrieving the frequency of each bigram.....	13
1.4.7 Saving to database.....	13
1.4.8 Predicting next word.....	13
1.4.9 Identifying errors.....	13
1.4.10 Finding the previous word to error.....	13
1.4.11 Predicting the correct word.....	14
CHAPTER 2. LITERATURE SURVEY.....	15
PAPER 1:Standardizing Tweets with Character-level Machine Translation.....	15
PAPER 2:Challenges of Multilingualism and Possible Approach for Standardization of e-Governance Solutions in India.....	17
PAPER 3:Rewriting the orthography of SMS messages.....	18
PAPER 4:Paralinguistic Restitution, Deletion, and Non-standard Orthography in text Messages.....	18

PAPER 5:Long Distance revision in Drafting and Post-Editing.....19

PAPER 6:Automatic standardization of texts containing spelling variation.....21

CHAPTER 3. SYSTEM DEVELOPMENT.....23

3.1 System Model.....23

3.2 System Design.....24

3.3 System Development.....25

CHAPTER 4. PERFORMANCE ANALYSIS.....42

4.1 Analysis.....42

4.2 Accuracy...../.....53

4.3 Precision.....53

4.4 Recall.....53

CHAPTER 5. CONCLUSION.....54

Conclusion.....54

Future works.....54

CHAPTER 6. REFERENCES.....55

LIST OF FIGURES

1. Methodology for supervised modeling.....	3
2. System Design flowchart.....	24

LIST OF GRAPHS:

- 1. Accessibility index due to Standardization.....17**
- 2. Three translation progression graphs showing distinction
is drafting, gisting and post editing.....19**
- 3. Two graphs Showing comparison in time taken by
students and professionals in drafting and post editing.....20**

ABSTRACT:

SMS are short-length text documents written in a informal style. SMS text processing is challenging because of multi-varied text composition in terms of language, vocabulary, style and quality. In this project, with the help of RapidMiner software tool we have tried to standardize SMS texts. We have worked on American English messages only. With the help of a slang dictionary, we corrected most of the word. In order to improve the efficiency of the system, we created a database to perform next word prediction from. We performed bigram on our corrected dataset, retrieved the previous values to the error, and from our prediction dataset predicted what possible words could be used. Our system gives an accuracy of about 96% and can be further improved.

1. INTRODUCTION:

1.1 INTRODUCTION:

The primary motivation for the creation and use of SMS language was to convey a comprehensible message using the fewest number of characters possible. This was for two reasons; one, telecommunication companies limited the number of characters per SMS, and also charged the user per SMS sent. To keep costs down, users had to find a way of being concise while still communicating the desired message. Two, typing on a phone is normally slower than with a keyboard, and capitalization is even slower. As a result, punctuation, grammar, and capitalization are largely ignored. In many countries, people now have access to unlimited text options in their monthly plan, although this varies widely from country to country, and operator to operator. However, screens are still small and the input problem persists, so SMS language is still widely used for brevity. Any word may be shortened (for example, "text" to "txt"). Words can also be combined with numbers to make them shorter (for example, "later" to "l8r"), using the numeral "8" for its homophonic quality.

Text standardization is rapidly gaining in popularity because of the explosion of user-generated text content in which language norms are not followed. SMS messages used to be the main object of text standardization while recently Twitter has started taking over as the most prominent source of information encoded with non-standard language.

There are two main approaches to text standardization.

1. **The unsupervised approach** mostly relies on phonetic transcription of non-standard words to produce standard candidates and language modeling on in-vocabulary (IV) data for selecting the most probable candidate.
2. **The supervised approach** assumes manually standardized data from which standardization models are built.

1. The unsupervised approach: There is no target variable is identified as such. Instead, the data mining algorithm searches for patterns and structure among all the variables. The most common unsupervised data mining method is clustering.

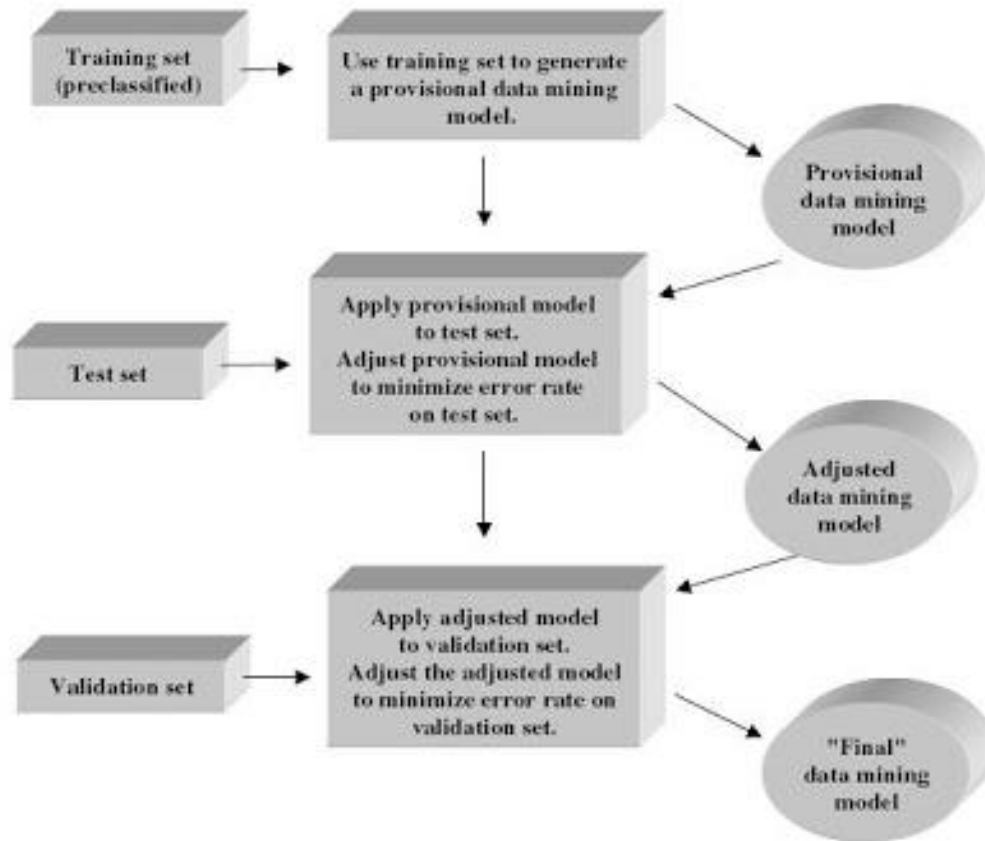
2. The supervised approach: Most data mining methods are supervised methods, however, meaning that (1) there is a particular pre-specified target variable, and (2) the algorithm is given many examples where the value of the target variable is provided, so

that the algorithm may learn which values of the target variable are associated with which values of the predictor variables.

Most supervised data mining methods apply the following methodology for building and evaluating a model.

First, the algorithm is provided with a training set of data, which includes the pre-classified values of the target variable in addition to the predictor variables. For example, if we are interested in classifying income bracket, based on age, gender, and occupation, our classification algorithm would need a large pool of records, containing complete (as complete as possible) information about every field, including the target field, income bracket. In other words, the records in the training set need to be pre-classified. A provisional data mining model is then constructed using the training samples provided in the training data set. However, the training set is necessarily incomplete; that is, it does not include the “new” or future data that the data modelers are really interested in classifying. Therefore, the algorithm needs to guard against “memorizing” the training set and blindly applying all patterns found in the training set to the future data. For example, it may happen that all customers named “David” in a training set may be in the high income bracket. We would presumably not want our final model, to be applied to new data, to include the pattern “If the customer’s first name is David, the customer has a high income.” Such a pattern is a spurious artifact of the training set and needs to be verified before deployment.

- i. The next step in supervised data mining methodology is to examine how the provisional data mining model performs on a test set of data. In the test set, a holdout data set, the values of the target variable are hidden temporarily from the provisional model, which then performs classification according to the patterns and structure it learned from the training set. The efficacies of the classifications are then evaluated by comparing them against the true values of the target variable.
- ii. The provisional data mining model is then adjusted to minimize the error rate on the test set.
- iii. The adjusted data mining model is then applied to a validation data set, another holdout data set, where the values of the target variable are again hidden temporarily from the model. The adjusted model is itself then adjusted, to minimize the error rate on the validation set. Estimates of model performance for future, unseen data can then be computed by observing various evaluative measures applied to the validation set.



Methodology for supervised modeling.

1.1.1 Natural Language Processing

Natural language processing (NLP) is a field of computer science, artificial intelligence, and computational linguistics concerned with the interactions between computers and human (natural) languages. As such, NLP is related to the area of human–computer interaction. Many challenges in NLP involve natural language understanding, that is, enabling computers to derive meaning from human or natural language input, and others involve natural language generation.

Modern NLP algorithms are based on machine learning, especially statistical machine learning. The paradigm of machine learning is different from that of most prior attempts at language processing. Prior implementations of language-processing tasks typically involved the direct hand coding of large sets of rules. The machine-learning paradigm

calls instead for using general learning algorithms — often, although not always, grounded in statistical inference — to automatically learn such rules through the analysis of large *corpora* of typical real-world examples. A *corpus* (plural, "corpora") is a set of documents (or sometimes, individual sentences) that have been hand-annotated with the correct values to be learned.

Typical applications for natural language processing include the following.

- A better human-computer interface that could convert from a natural language into a computer language and vice versa. A natural language system could be the interface to a database system, such as for a travel agent to use in making reservations. Blind people could use a natural language system (with speech recognition) to interact with computers, and Steven Hawking uses one to generate speech from his typed text.
- A translation program that could translate from one human language to another (English to French, for example). Even if programs that translate between human languages are not perfect, they would still be useful in that they could do the rudimentary translation first, with their work checks and corrected by a human translator. This cuts down on the time for the translation.
- Programs that could check for grammar and writing techniques in a word processing document.
- A computer that could read a human language could read whole books to stock its database with data.

Major tasks of NLP:

The following is a list of some of the most commonly researched tasks in NLP. Note that some of these tasks have direct real-world applications, while others more commonly serve as subtasks that are used to aid in solving larger tasks. What distinguishes these tasks from other potential and actual NLP tasks is not only the volume of research devoted to them but the fact that for each one there is typically a well-defined problem setting, a standard metric for evaluating the task, standard corpora on which the task can be evaluated, and competitions devoted to the specific task.

1. **Automatic summarization**: Produce a readable summary of a chunk of text. Often used to provide summaries of text of a known type, such as articles in the financial section of a newspaper.
2. **Coreference resolution**: Given a sentence or larger chunk of text, determine which words ("mentions") refer to the same objects ("entities"). Anaphora resolution is a specific example of this task, and is specifically concerned with

matching up pronouns with the nouns or names that they refer to. The more general task of co-reference resolution also includes identifying so-called "bridging relationships" involving referring expressions. For example, in a sentence such as "He entered John's house through the front door", "the front door" is a referring expression and the bridging relationship to be identified is the fact that the door being referred to is the front door of John's house (rather than of some other structure that might also be referred to).

3. **Discourse analysis**: This rubric includes a number of related tasks. One task is identifying the discourse structure of connected text, i.e. the nature of the discourse relationships between sentences (e.g. elaboration, explanation, contrast). Another possible task is recognizing and classifying the speech acts in a chunk of text (e.g. yes-no question, content question, statement, assertion, etc.).
4. **Machine translation**: Automatically translate text from one human language to another. This is one of the most difficult problems, and is a member of a class of problems colloquially termed "AI-complete", i.e. requiring all of the different types of knowledge that humans possess (grammar, semantics, facts about the real world, etc.) in order to solve properly.
5. **Morphological segmentation**: Separate words into individual morphemes and identify the class of the morphemes. The difficulty of this task depends greatly on the complexity of the morphology (i.e. the structure of words) of the language being considered. English has fairly simple morphology, especially inflectional morphology, and thus it is often possible to ignore this task entirely and simply model all possible forms of a word (e.g. "open, opens, opened, opening") as separate words. In languages such as Turkish or Manipuri, a highly agglutinated Indian language, however, such an approach is not possible, as each dictionary entry has thousands of possible word forms.
6. **Named entity recognition (NER)**: Given a stream of text, determine which items in the text map to proper names, such as people or places, and what the type of each such name is (e.g. person, location, organization). Note that, although capitalization can aid in recognizing named entities in languages such as

English, this information cannot aid in determining the type of named entity, and in any case is often inaccurate or insufficient. For example, the first word of a sentence is also capitalized, and named entities often span several words, only some of which are capitalized. Furthermore, many other languages in non-Western scripts (e.g. Chinese or Arabic) do not have any capitalization at all, and even languages with capitalization may not consistently use it to distinguish names. For example, German capitalizes all nouns, regardless of whether they refer to names, and French and Spanish do not capitalize names that serve as adjectives.

7. **Natural language generation**: Convert information from computer databases into readable human language.
8. **Natural language understanding**: Convert chunks of text into more formal representations such as first-order logic structures that are easier for computer programs to manipulate. Natural language understanding involves the identification of the intended semantic from the multiple possible semantics which can be derived from a natural language expression which usually takes the form of organized notations of natural languages concepts. Introduction and creation of language meta-model and ontology are efficient however empirical solutions. An explicit formalization of natural languages semantics without confusions with implicit assumptions such as closed-world assumption (CWA) vs. open-world assumption, or subjective Yes/No vs. objective True/False is expected for the construction of a basis of semantics formalization.
9. **Optical character recognition (OCR)**: Given an image representing printed text, determine the corresponding text.
10. **Part-of-speech tagging**: Given a sentence, determine the part of speech for each word. Many words, especially common ones, can serve as multiple parts of speech. For example, "book" can be a noun ("the book on the table") or verb ("to book a flight"); "set" can be a noun, verb or adjective; and "out" can be any of at least five different parts of speech. Some languages have more such ambiguity than others. Languages with little inflectional morphology, such as English are

particularly prone to such ambiguity. Chinese is prone to such ambiguity because it is a tonal language during verbalization. Such inflection is not readily conveyed via the entities employed within the orthography to convey intended meaning.

- 11. Parsing:** Determine the parse tree (grammatical analysis) of a given sentence. The grammar for natural languages is ambiguous and typical sentences have multiple possible analyses. In fact, perhaps surprisingly, for a typical sentence there may be thousands of potential parses (most of which will seem completely nonsensical to a human).
- 12. Question answering:** Given a human-language question, determine its answer. Typical questions have a specific right answer (such as "What is the capital of Canada?"), but sometimes open-ended questions are also considered (such as "What is the meaning of life?"). Recent works have looked at even more complex questions.
- 13. Relationship extraction:** Given a chunk of text, identify the relationships among named entities (e.g. who is married to whom).
- 14. Sentence breaking (also known as sentence boundary disambiguation):** Given a chunk of text, find the sentence boundaries. Sentence boundaries are often marked by periods or other punctuation marks, but these same characters can serve other purposes (e.g. marking abbreviations).
- 15. Sentiment analysis:** Extract subjective information usually from a set of documents, often using online reviews to determine "polarity" about specific objects. It is especially useful for identifying trends of public opinion in the social media, for the purpose of marketing.
- 16. Speech recognition:** Given a sound clip of a person or people speaking, determine the textual representation of the speech. This is the opposite of text to speech and is one of the extremely difficult problems colloquially termed "AI-complete" (see above). In natural speech there are hardly any pauses between successive words, and thus speech segmentation is a necessary subtask of speech

recognition (see below). Note also that in most spoken languages, the sounds representing successive letters blend into each other in a process termed co-articulation, so the conversion of the analog signal to discrete characters can be a very difficult process.

17. **Speech segmentation**: Given a sound clip of a person or people speaking, separate it into words. A subtask of speech recognition and typically grouped with it.
18. **Topic segmentation and recognition**: Given a chunk of text, separate it into segments each of which is devoted to a topic, and identify the topic of the segment.
19. **Word segmentation**: Separate a chunk of continuous text into separate words. For a language like English, this is fairly trivial, since words are usually separated by spaces. However, some written languages like Chinese, Japanese and Thai do not mark word boundaries in such a fashion, and in those languages text segmentation is a significant task requiring knowledge of the vocabulary and morphology of words in the language.
20. **Word sense disambiguation**: Many words have more than one meaning; we have to select the meaning which makes the most sense in context. For this problem, we are typically given a list of words and associated word senses, e.g. from a dictionary or from an online resource such as WordNet.

In some cases, sets of related tasks are grouped into subfields of NLP that are often considered separately from NLP as a whole. Examples include:

- i. **Information retrieval (IR)**: This is concerned with storing, searching and retrieving information. It is a separate field within computer science (closer to databases), but IR relies on some NLP methods (for example, stemming). Some current research and applications seek to bridge the gap between IR and NLP.
- ii. **Information extraction (IE)**: This is concerned in general with the extraction of semantic information from text. This covers tasks such

as named entity recognition, Co-reference resolution, relationship extraction, etc.

- iii. **Speech processing**: This covers speech recognition, text-to-speech and related tasks.

The tool we are using for this project is **RAPIDMINER**

1.1.2 Software Information: RapidMiner

RapidMiner is a software platform developed by the company of the same name that provides an integrated environment for machine learning, data mining, text mining, predictive analytics and business analytics. It is used for business and industrial applications as well as for research, education, training, rapid prototyping, and application development and supports all steps of the data mining process including results visualization, validation and optimization.

RapidMiner provides 99% of an advanced analytical solution through template-based frameworks that speed delivery and reduce errors by nearly eliminating the need to write code. RapidMiner provides data mining and machine learning procedures including: data loading and transformation (Extract, transform, load (ETL)), data preprocessing and visualization, predictive analytics and statistical modeling, evaluation, and deployment. RapidMiner is written in the Java programming language. RapidMiner provides a GUI to design and execute analytical workflows. Those workflows are called “Process” in RapidMiner and they consist of multiple “Operators”.

Each operator is performing a single task within the process and the output of each operator forms the input of the next one. Alternatively, the engine can be called from other programs or used as an API. Individual functions can be called from the command line. RapidMiner provides learning schemes and models and algorithms from Weka and Rscripts that can be used through extensions.

Text Mining: Text mining (also referred to as text data mining or knowledge discovery from textual databases), refers to the process of discovering interesting and non-trivial knowledge from text documents. The common practice in text mining is the analysis of the information extracted through text processing to form new facts and new hypotheses that can be explored further with other data mining algorithms. Text mining applications typically deal with large and complex data sets of textual documents that contain significant amount of irrelevant and noisy information. Feature selection aims to remove this irrelevant and noisy information by focusing only on relevant and informative data for use in text mining. Some of the topics within text mining include feature extraction, text categorization, clustering, trends analysis, association mining and visualization.

Text processing is an important extension to perform text mining. The Text Extension adds all operators necessary for statistical text analysis. You can load texts from many different data sources, transform them by a huge set of different filtering techniques, and finally analyze your text data.

The Text Extensions supports several text formats including plain text, HTML, or PDF as well as other data sources. It provides standard filters for tokenization, stemming, stopword filtering, or n-gram generation to provide everything necessary for preparing and analyzing texts.

1.2 PROBLEM STATEMENT:

To design a system that is able to identify and standardize macronic or multilingual language in the data provided to the system.

In the modern world, due to various reasons, people have started using short hand, texting language like slangs etc, which calls for the need of text standardization. Also now the people have started mixing two or more language in the same text. We aim to standardize these texts (in specific languages) to a standard language.

1.3 OBJECTIVE:

Our objective is to achieve:

1. To find the dataset of the language which has the same script
2. To learn Rapid Miner
3. To handle macaronic text by auto identifying the text
4. To predict the next word

1.4 METHODOLOGY:

Our methodology can be divided into basically 9 modules:

1. Identifying and Collecting Dataset
2. Filtering
3. Tokenization
4. Standardization
5. Generating bi-grams
6. Retrieving the frequency of each bigram
7. Saving to database
8. Predicting next word
9. Identifying errors
10. Finding the previous word to error
11. Predicting the correct word

1.4.1 Identifying and Collecting Dataset:

In this module, the dataset from various sources is collected. It is important to have an understanding and idea about the data set to be worked on. We need to be certain about the data we would be working on.

1.4.2 Filtering:

In our project, filtering in different phases is involved. Filtering enables us to sort out the data we have collected and help us to work more efficiently. Initially we conduct filtering manually, where we decide on what data to precisely work on. This is done to remove

unnecessary data, if any, from the data we have collected. Then, once tokenization is performed, the data is again filtered in order to get the precise information we need to work on.

1.4.3 Tokenization:

Tokenization is the act of breaking up a sequence of strings into pieces such as words, keywords, phrases, symbols and other elements called tokens. Tokens can be individual words, phrases or even whole sentences. In the process of tokenization, some characters like punctuation marks are discarded. The tokens become the input for another process like parsing and text mining.

Tokenization is used in computer science, where it plays a large part in the process of lexical analysis.

1.4.4 Standardization:

In this we standardize our text into a uniform language and get our desired results. It is performed manually as well as automatically. For this project we took 3 users to manually standardize the text we provided and write their slangs and the correct word. This also helped us to identify how well our automated system is working.

It is important for our data to be digital for it to be processed in an automated system. If the source is already digital text (such as a file in text, XML, or HTML format), or you have converted the source to that, your next task is to make sure that the text complies

After this process we get our desired result.

1.4.5 Generating bi-grams:

An n-gram model is a type of probabilistic language model for predicting the next item in such a sequence in the form of a $(n - 1)$ -order Markov model. An n-gram model models sequence, notably natural languages, using the statistical properties of n-grams.

An n-gram model predicts x_i based on $x_{i-(n-1)}, \dots, x_{i-1}$. In probability terms, this is $P(x_i | x_{i-(n-1)}, \dots, x_{i-1})$. When used for language modeling, independence assumptions are made so that each word depends only on the last $n - 1$ words. This Markov model is used as an approximation of the true underlying language. This assumption is important because it massively simplifies the problem of learning the language model from data.

By generating bigrams, we can determine the combinations of words occurring in our corpus, and this information, hence helps us determine the word which must follow.

1.4.6 Retrieving the frequency of each bigram:

After we have attained our data, after generating bi-grams, we determine the term occurrences. We can now determine the frequency of each combination occurring in our corpus.

This will enable us to know what combination has occurred the most, therefore, helping us determine what word should follow.

1.4.7 Saving to database:

In this, once we have attained the frequency of each bigram and the pair of words, we save this data to database so that it can be fetched and the next word prediction can be performed.

1.4.8 Predicting the next word:

Once our database is created, we retrieve the next word form it based on certain conditions and successfully retrieve our output.

1.4.9 Identifying the error:

We use our prediction system to correct the errors of our initial system. It is possible that the dictionary doesn't contain some errors. So using our prediction system, we can find out the correct word.

1.4.10 Finding the previous word to error:

We find out the previous word to error in order to compare it with our prediction database. This would enable us to know what word we must look for in our database. This can be done using bigram on our corpus.

1.4.11 Predicting the correct word:

Once we find our previous word, we look at the words corresponding to that word, giving us possible solution. The user can then see the correct word and know what should be the correct word.

2. LITERATURE STUDY

PAPER 1:

Standardizing Tweets with Character-level Machine Translation

Abstract

This paper presents the results of the standardization procedure of Slovene tweets that are full of colloquial, dialectal and foreign-language elements. With the aim of minimizing the human input required we produced a manually normalized lexicon of the most salient out-of-vocabulary (OOV) tokens and used it to train a character-level statistical machine translation system (CSMT). Best results were obtained by combining the manually constructed lexicon and CSMT as fallback with an overall improvement of 9.9% increase on all tokens and 31.3% on OOV tokens.

Manual preparation of data in a lexicon manner has proven to be more efficient than normalizing running text for the task at hand. Finally we performed an extrinsic evaluation where we automatically lemmatized the test corpus taking as input either original or automatically standardized word forms, and achieved 75.1% per-token accuracy with the former and 83.6% with the latter, thus demonstrating that standardization has significant benefits for upstream processing.

Dataset Collection

The basis for our dataset was the database of tweets from the now no longer active aggregator sitweet.com containing (mostly) Slovene tweets posted between 2007-01-12 and 2011-02-20. The database contains many tweets in other languages as well, so we first used a simple filter that keeps only those that contain one of the Slovene letters. This does not mean that there is no foreign language text remaining, as some closely related languages, in particular Croatian, also use these letters.

Also it is fairly common to mix Slovene and another language, mostly English, in a single tweet. However, standard methods for language identification do not work well with the type of language found in tweets, and are also bad at distinguishing closely related languages, especially if a single text uses more than one language. In this step we

also shuffled the tweets in the collection so that taking any slice will give a random selection of tweets, making it easier to construct training and testing datasets.

In the second step we anonymized the tweets by substituting hashtags, mentions and URLs with special symbols (XXX-HST, XXX-MNT, XXX-URL) and substituted emoticons with XXX-EMO. This is meant to serve two purposes. On the one hand, we make the experimental dataset freely available and by using rather old and anonymized tweets we hope to evade problems with

the Twitter terms of use. On the other, tweets are difficult to tokenize correctly and by substituting symbols for the most problematic tokens, i.e. emoticons, we made the collection easier to process.

We then tokenized the collection and stored it in the so called vertical format, where each line is either an XML tag (in particular, <text> for an individual (tweet) or one token. With this we obtained a corpus of about half a million tweets and eight million word tokens which is the basis for our datasets.

Experiments and results

Our overall approach to tweet standardization is based on standardizing only OOV tokens by applying transformations on them with the goal of producing wordforms identical to the ones produced during manual corpus standardization.

Therefore we evaluate our approaches with two types of accuracy on the corpus:

1. ACC-ALL { accuracy on all word tokens in the corpus
2. ACC-OOV { accuracy on OOV word tokens in the corpus

The first measure reports how well we do on the level of complete texts, and the second one how well we do on the tokens we perform our transformations on .We perform all together five sets of experiments.

- CSMT datasets
- Lower and upper bounds
- CSMT extensions
- Lexicon vs. corpus standardization
- Lemmatization experiment

PAPER 2:**Challenges of Multilingualism and Possible Approach for Standardization of e-Governance Solutions in India****Abstract**

In this paper we have addressed the major challenges and issues involved in the multilingualism aspects towards standardization of e-governance solutions in India. The paper also investigates the benefits of adopting open standards and open source software in implementing multilingual e-governance solutions.

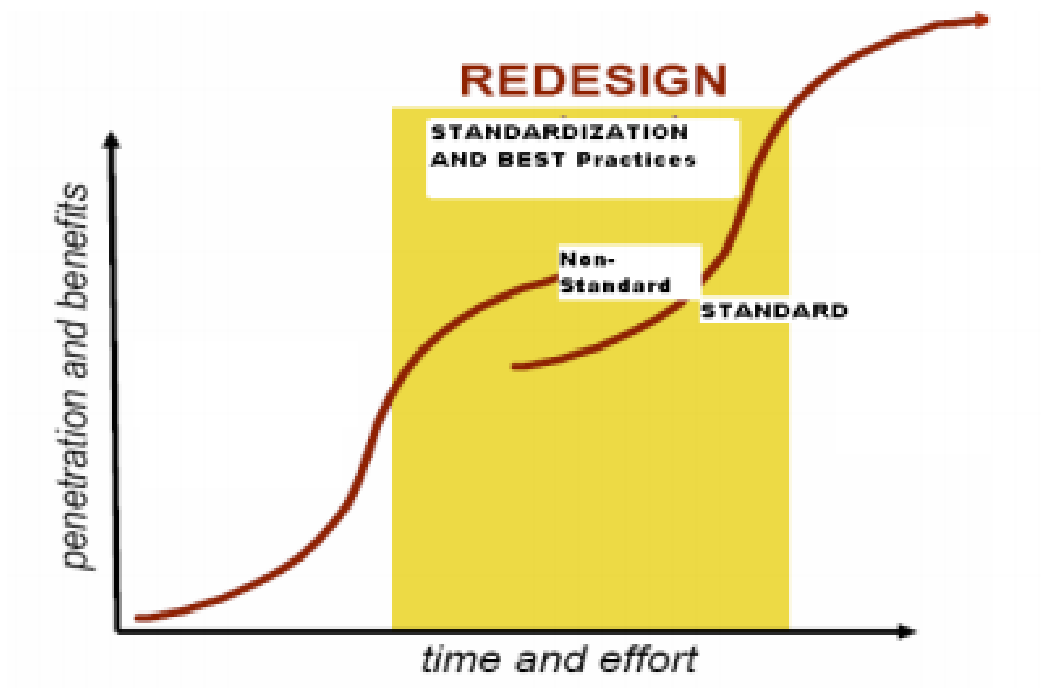


Fig.0 Accessibility index due to Standardization

PAPER 3:**Rewriting the orthography of SMS messages****Abstract**

Electronic written texts used in computer-mediated interactions (emails, blogs, chats, and the like) contain significant deviations from the norm of the language. This paper presents the detail of a system aiming at normalizing the orthography of French SMS messages: after discussing the linguistic peculiarities of these messages and possible approaches to their automatic normalization, we present, compare, and evaluate various instantiations of a normalization device based on weighted finite-state transducers. These experiments show that using an intermediate phonemic representation and training, our system outperforms an alternative normalization system based on phrase-based statistical machine translation techniques.

PAPER 4:**Paralinguistic Restitution, Deletion, and Non-standard Orthography in Text Messages****Abstract**

This thesis examines the structure of text messages. In recent years, literature speculating about electronically mediated communication has proliferated. An abundance of literature on technology and language exists, but little of it explores text messaging. The literature that looks at texting tends to focus on the social aspects of text communication or on the damage people fear it will cause to language. Little literature focuses on empirical analysis of text messaging from a linguistic perspective. Text messages are a communication medium with limitations and intricacies all their own, and they deserve attention. The informal nature of texting allows for a variety of lexical and grammatical creativity. Letter and word deletions appear, perhaps inspired by the 160 character per message length limit. Unconventional punctuation and spelling abound. Text messaging has become a significant part of language use in our culture, especially for young people. Today, phones are used more for text messaging than for voice communication in many countries. Texting is a vital piece of the technology-mediated-communication puzzle and warrants inspection; we cannot tackle the question of what digital technology means for social interaction or for language until we understand text messaging structurally. It is worth remembering, too, that as phones and phone plans advance, the 160 character limit - one of the factors unique to texting and perhaps integral in generating the new linguistic

phenomena we see in text speak - will become less meaningful. Perhaps even more critically, texting patterns are also changing as phones change. In analyzing the structure of Elizondo, 2 text messaging today, we may be capturing a unique moment in the tech-language trajectory before a new type of electronically mediated communication replaces or changes texting and we lose this piece of the language history.

PAPER 5:

Long Distance revision in Drafting and Post-Editing

Abstract

This paper investigates properties of translation processes, as observed in the translation behaviour of student and professional translators. The translation process can be divided into a gisting, drafting and post-editing phase. We find that student translators have longer gisting phases whereas professional translators have longer post-editing phases. Long-distance revisions, which would typically be expected during post-editing, occur to the same extent during drafting as during post-editing. Further, both groups of translators seem to face the same translation problems. We suggest how those findings might be taken into account in the design of computer assisted translation tools.

Some figures:

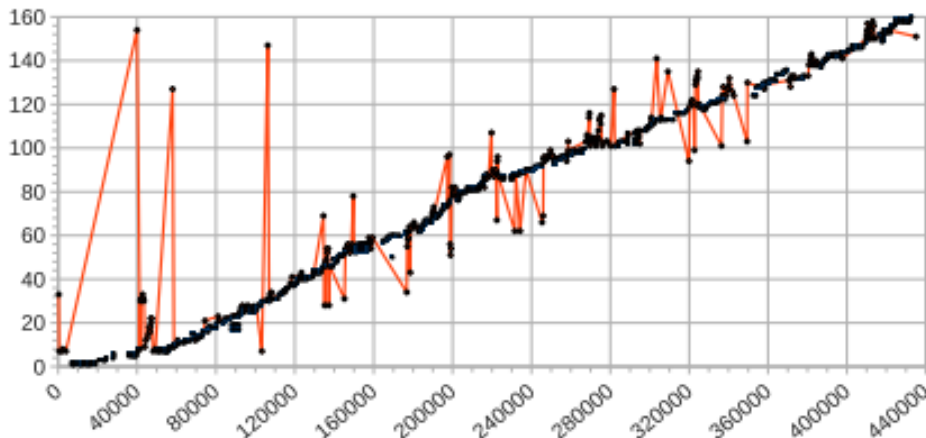
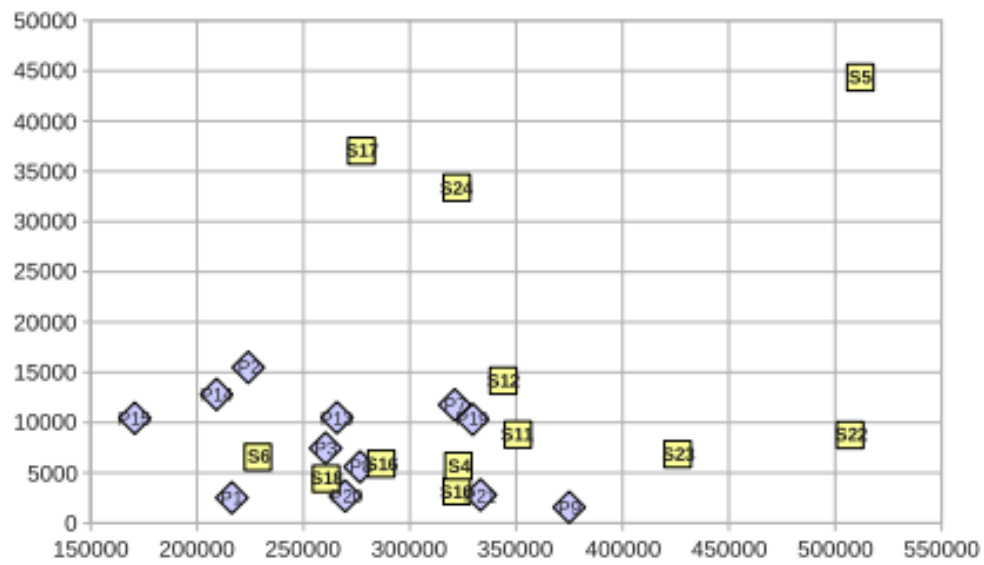
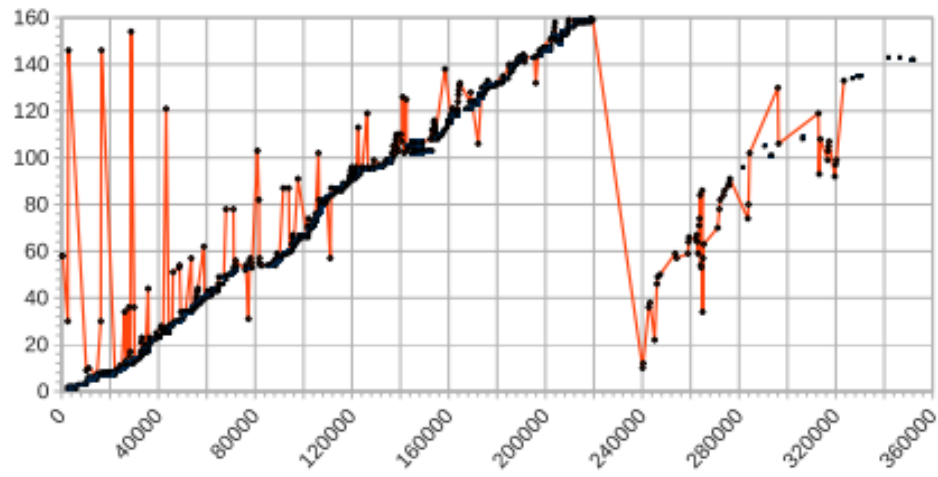
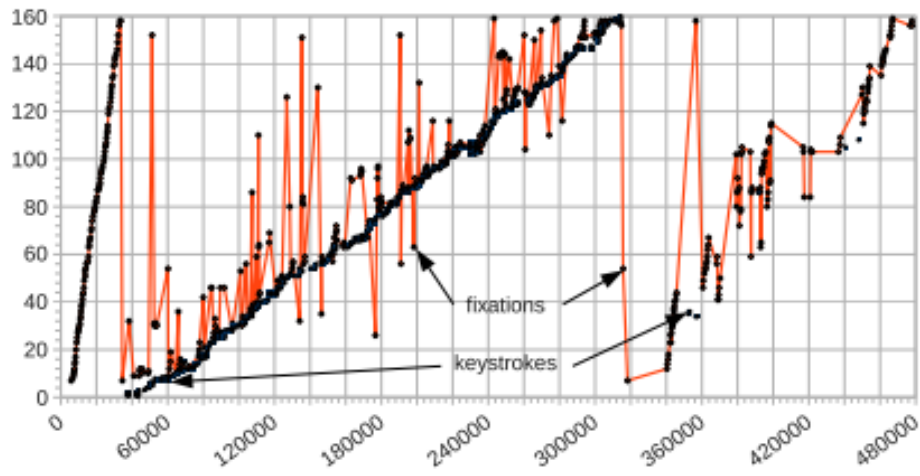


Fig. 1. Three translation progression graphs from top down subjects S17, P1 and S23, showing keystrokes and eye movements: S17 shows a clear distinction into gisting, drafting and post-editing. P1 has no gisting phase and spends almost 50% of the translation time on post-editing, while S23 only has a drafting phase.



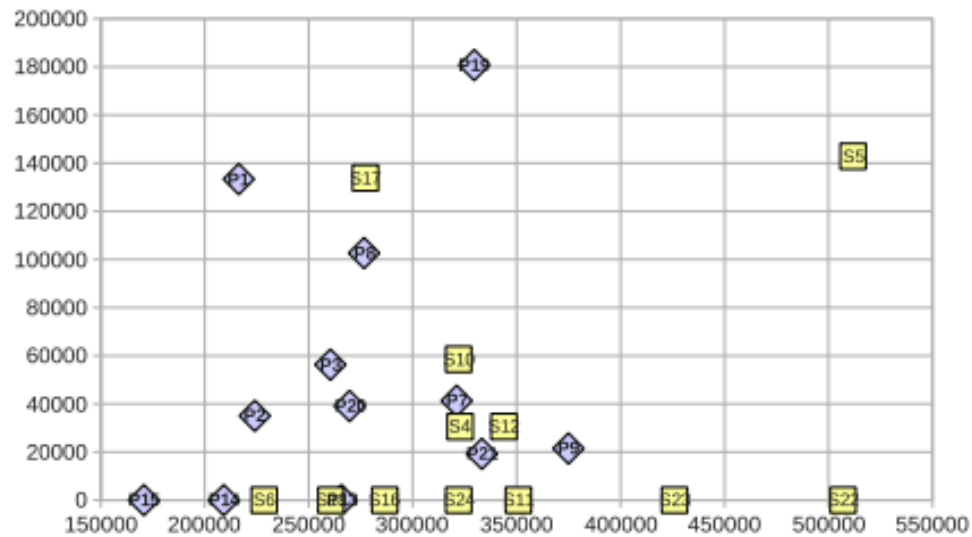


Fig. 2. Top: drafting time (horizontal) and gisting time (vertical). Rectangular symbols represent student translators, diamond shapes represent professionals. Students spend more time on gisting than professionals. Bottom: drafting time (horizontal) and post-editing time (vertical). Rectangular symbols represent students, diamond shapes represent professionals. On average, professionals spend more time post-editing than do students; many students completely skip post-editing.

PAPER 6:

Automatic standardization of texts containing spelling variation

Abstract

Large quantities of spelling variation in corpora, such as that found in Early Modern English, can cause significant problems for corpus linguistic tools and methods. Having texts with standardized spelling is key to making such tools and methods accurate and meaningful in their analysis. Gaining access to such versions of texts can be problematic however, and manual standardization of the texts is often too time-consuming to be feasible.

Our solution is a piece of software named VARD 2 which can be used to manually and automatically standardize spelling variation in individual texts, or corpora of any size. This paper evaluates VARD 2's performance on a corpus of Early Modern English letters and a corpus of children's written English. The software's ability to learn from manual standardization is put under particular scrutiny as we examine what effect different levels of training have on its performance.

3. SYSTEM DEVELOPMENT

3.1 SYSTEM MODEL:

In this project we use the incremental model.

The incremental build model is a method of software development where the product is designed, implemented and tested incrementally a little more is added each time until the product is finished

In incremental model the whole requirement is divided into various phases. Multiple development cycles take place making the life cycle a “multi-waterfall” cycle. Cycles are divided up into smaller, more easily managed modules. Each module passes through the requirements, design, implementation and testing phases. A working version of software is produced during the first module, so we get working software early on during the software life cycle. Each subsequent release of the module adds function to the previous release. The process continues till the complete system is achieved.

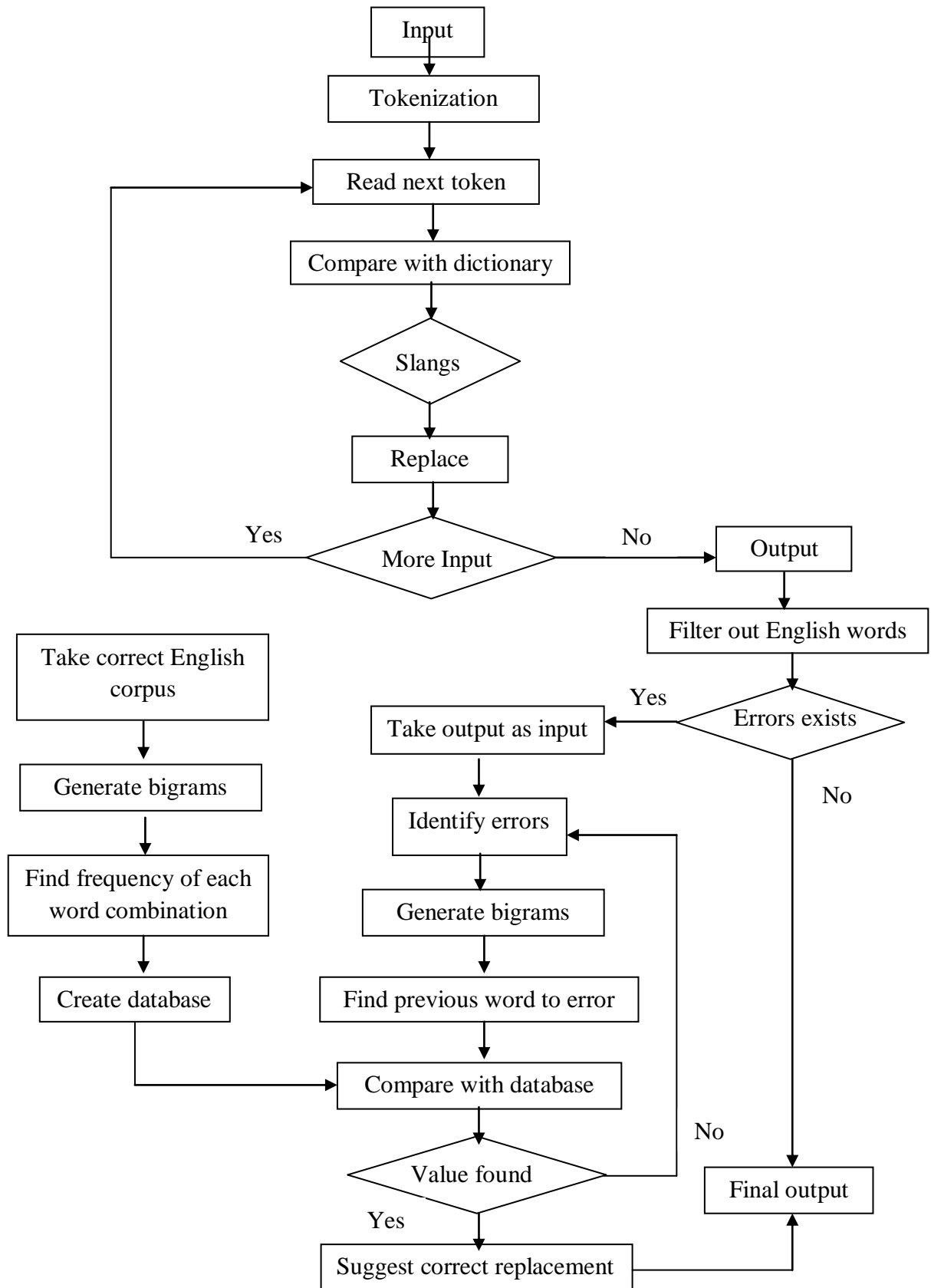
- This model can be used when the requirements of the complete system are clearly defined and understood.
- Major requirements must be defined whereas; some details can evolve with time.
- There are some high risk features and goals.

In this project, in the first phase words are standardized with respect to a “slang dictionary.

In second phase, we create database for predicting the correct word.

In the third phase, we sync our corrected data, identify the wrong words, and predict the correct word from our prediction database.

3.2 SYSTEM DESIGN:



3.3 SYSTEM DEVELOPMENT:

The following step were undertaken to get our desired output from our system:

1. Collection of data:

We collected data from various sources and sorted out and finalized around 70 sms, statuses etc to work on for our initial stage.

Various data sources:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?><smsCorpus date="2015.03.09" version="1.2">
<message id="10120"><text>Bugis oso near wat...</text><source><srcNumber>51</srcNumber><phoneModel manufacturer="unknown" smartphone="unknown"/><us
<message id="10121"><text>Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat...</text><s
<message id="10122"><text>I dunno until when... Lets go learn pilates...</text><source><srcNumber>51</srcNumber><phoneModel manufacturer="unknown"
<message id="10123"><text>Den only weekdays got special price... Haiz... Cant eat liao... Cut nails oso muz wait until i finish drivin wat, lunch
<message id="10124"><text>Meet after lunch la...</text><source><srcNumber>51</srcNumber><phoneModel manufacturer="unknown" smartphone="unknown"/><u
<message id="10125"><text>m walking in citylink now ü faster come down... Me very hungry...</text><source><srcNumber>51</srcNumber><phoneModel man
<message id="10126"><text>5 nights...We nt staying at port step liao...Too ex</text><source><srcNumber>51</srcNumber><phoneModel manufacturer="unkn
<message id="10127"><text>Hey pple...$700 or $900 for 5 nights...Excellent location wif breakfast hamper!!!</text><source><srcNumber>51</srcNumber
<message id="10128"><text>Yun ah.the ubi one say if ü wan call by tomorrow.call 67441233 look for irene.ere only got bus8,22,65,61,66,382. Ubi cre
<message id="10129"><text>Hey tmr maybe can meet you at yck</text><source><srcNumber>51</srcNumber><phoneModel manufacturer="unknown" smartphone="u
<message id="10130"><text>Oh...i asked for fun. Haha...take care. ü</text><source><srcNumber>51</srcNumber><phoneModel manufacturer="unknown" smart
<message id="10131"><text>We are supposed to meet to discuss abt our trip... Thought xuhui told you? In the afternoon. Thought we can go for lessa
<message id="10132"><text>t finish my film yet...</text><source><srcNumber>51</srcNumber><phoneModel manufacturer="unknown" smartphone="unknown"/><
<message id="10133"><text>m having dinner with my cousin...</text><source><srcNumber>51</srcNumber><phoneModel manufacturer="unknown" smartphone="u
<message id="10134"><text>Oh... Kay... On sat right?</text><source><srcNumber>51</srcNumber><phoneModel manufacturer="unknown" smartphone="unknown"
<message id="10135"><text>I need... Coz i never go before</text><source><srcNumber>51</srcNumber><phoneModel manufacturer="unknown" smartphone="unk
<message id="10136"><text>s a basic yoga course... at bugis... We can go for that... Pilates intro next sat.... Tell me what time you r free</text
<message id="10137"><text>I am going to sao mu today. Will be done only at 12</text><source><srcNumber>51</srcNumber><phoneModel manufacturer="unkn
<message id="10138"><text>Hey gals...U all wanna meet 4 dinner at nite?</text><source><srcNumber>51</srcNumber><phoneModel manufacturer="unknown" s
<message id="10139"><text>Jos ask if u wana meet up?</text><source><srcNumber>51</srcNumber><phoneModel manufacturer="unknown" smartphone="unknown"
<message id="10140"><text>Haiyoh... Maybe your hamster was jealous of million</text><source><srcNumber>51</srcNumber><phoneModel manufacturer="unkn
<message id="10141"><text>is your hamster dead? Hey so tmr i meet you at 1pm orchard mrt?</text><source><srcNumber>51</srcNumber><phoneModel manuf
<message id="10142"><text>ve booked the pilates and yoga lesson already... Haha</text><source><srcNumber>51</srcNumber><phoneModel manufacturer="un
<message id="10143"><text>Yup... I havent been there before... You want to go for the yoga? I can call up to book</text><source><srcNumber>51</src
<message id="10144"><text>K... Must book a not huh? so going for yoga basic on sunday?</text><source><srcNumber>51</srcNumber><phoneModel manufact
<message id="10145"><text>Hey so this sat are we going for the intro pilates only? Or the kickboxing too?</text><source><srcNumber>51</srcNumber><
<message id="10146"><text>Sat right? Okay thanks...</text><source><srcNumber>51</srcNumber><phoneModel manufacturer="unknown" smartphone="unknown"/
<message id="10147"><text>Yup... From what i remb... I think should be can book...</text><source><srcNumber>51</srcNumber><phoneModel manufacturer=
<message id="10148"><text>m going to get specs. My membership is PX3748</text><source><srcNumber>51</srcNumber><phoneModel manufacturer="unknown" s
<message id="10149"><text>We can go 4 e normal pilates after our intro...</text><source><srcNumber>51</srcNumber><phoneModel manufacturer="unknown"
<message id="10150"><text>Sun ah... Thk mayb can if dun have anythin on... Thk have to book e lesson... E pilates is at orchard mrt u noe hor...</
<message id="10151"><text>Thk shld h can Ya i wana go 4 lessons Haha can go for one whole stretch </text><source><srcNumber>51</srcNumber
```

'Go until jurong point, crazy... Available only in bugis n great world la e buffet... Cine there got amore wat...'.0
 'Ok lar... Joking wif u oni...'.0
 'Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question(std txt rate)T&C's apply 08452810075over18's',1
 'U dun say so early hor... U c already then say...'.0
 'Nah I don't think he goes to usf, he lives around here though'.0
 'FreeMsg Hey there darling it's been 3 week's now and no word back! I'd like some fun you up for it still? Tb ok! XxX std chgs to send, Â£1.50 to rcv'.1
 'Even my brother is not like to speak with me. They treat me like aids patent.'.0
 'As per your request \'Melle Melle (Oru Minnaminunginte Nurungu Vettam)\' has been set as your callertune for all Callers. Press *9 to copy your friends Callertune'.0
 'WINNER!! As a valued network customer you have been selected to receive a Â£900 prize reward! To claim call 09061701461. Claim code KL341. Valid 12 hours only.'.1
 'Had your mobile 11 months or more? U R entitled to Update to the latest colour mobiles with camera for Free! Call The Mobile Update Co FREE on 08002986030'.1
 'I\'m gonna be home soon and i don't want to talk about this stuff anymore tonight, k? I've cried enough today.'.0
 'SIX chances to win CASH! From 100 to 20,000 pounds txt> CSH11 and send to 87575. Cost 150p/day, 6days, 16+ TsandCs apply Reply HL 4 info'.1
 'URGENT! You have won a 1 week FREE membership in our Â£100,000 Prize Jackpot! Txt the word: CLAIM to No: 81010 T&C www.dbuk.net LCCLTD POBOX 4403LDNW1A7RW18'.1
 'I\'ve been searching for the right words to thank you for this breather. I promise i wont take your help for granted and will fulfil my promise. You have been wonderful and a blessing at all times.'.0
 'I HAVE A DATE ON SUNDAY WITH WILL!!'.0

Final data selected:

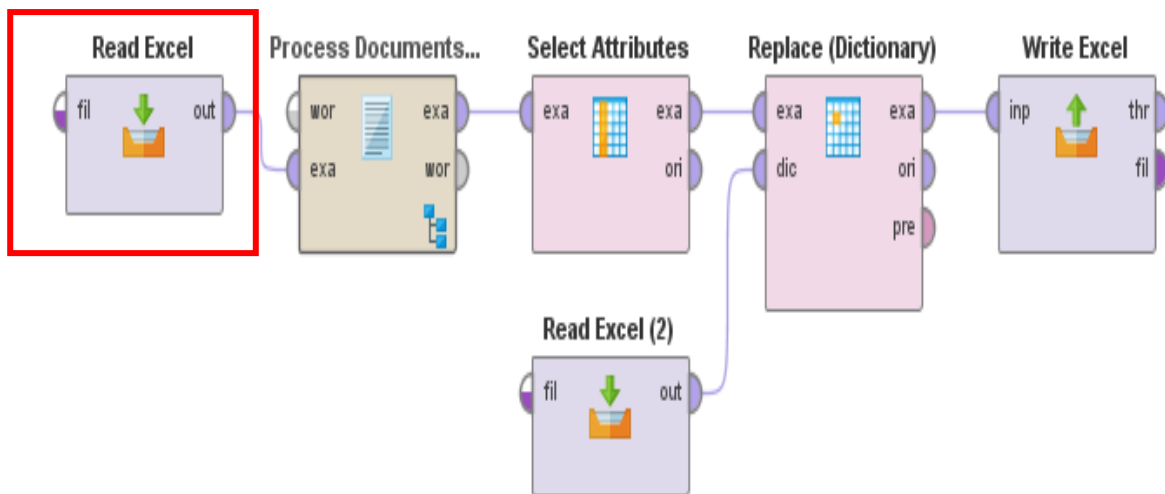
Id	Message
1	m nt goin, got somethin on
2	i dun mind goin jazz oso.
3	goin 4 my drivin den go shoppin after tt
4	Cut nails oso muz wait until i finish drivin wat, lunch still muz eat wat...
5	Den wat will e schedule b lk on Sun?
6	m not at home in da aftn wat.
7	m on da bus going home.
8	m eatin wif my frens now
9	Do u noe how 2 send files between two computers?
10	U meet other fren dun wan meet me ah... Muz b a guy rite.
11	Dunno i shd b driving or not cos i go sch 1 hr oni.
12	hv to finish my film yet...
13	I jokin oni
14	Oh... Kay... On Sat right?
15	Wat r u up 2 ?
16	lol... ryt thr.

17	u r so sweet... i like u so much, but u seem to be having attitude...
18	Hpe we will meet soon.
19	Dis is d ryt choice
20	He ws awesome
21	God,I jst lk her
22	I hv 2 b dere
23	wat did u do?
24	Try 2 smyl
25	f he cnt fnd a resn, he shud go to authorities
26	Ard 6 like dat
27	But i hv enuff space got like 4mb
28	Where r we meeting tmr?
29	Il b going 2 sch on Fri quite early lor cos mys sis got paper in da morn
30	Il ttlyl, bzy ryt nw
31	wat u wana do?
32	All e best 4 ur driving tmr
33	Ok but tell me half an hr b4 u come i need 2 prepare
34	Yar... I tot u knew dis would happen long ago already.
35	Ok... Take ur time n enjoy ur dinner...
36	Yup... We r at 2nd floor boots...
37	u reach home den let me noe
38	U gt to tl dem its nt ryt.
39	stp laughin at me.
40	hy, sry cnt meet u today.
41	im almst dne.
42	Nope din buy anythin.
43	muz b still enjoyin herself
44	whr hv u been?
45	My sis will drive me dere
46	I oso wont stay 4 too long
47	Wat time is ur tuition?
48	V to meet em at 845
49	m comin bk nw.
50	No biggy.
51	r hols included?
52	Aftr a week il cm hme.
53	mom is on her way.
54	Prepared fr exams?

55	c d intruder
56	m comin soon, gota wait 4 my sis first
57	m in d car ... Cant send u.
58	Thanx 4 d gift...Muz tell me when u free
59	Il b present nxt Mon. Den was wonderin how come not on Fri
60	Got pts one huh... U joinin, if u r i dun mind
61	Il mt u all too.
62	idk wat u sayin.
63	brb. I gtg.
64	he's nt angry w her bf since he came to pick her yest.
65	Il msg u to kp u informed
66	But u muz tell me wat u wan to noe
67	Hope u find wot was lost
69	did u c my aftn msg?
70	hv nt tot abt it

2. Importing the data:

Once we had our final data to work on, we loaded this data on our software by reading the excel file.



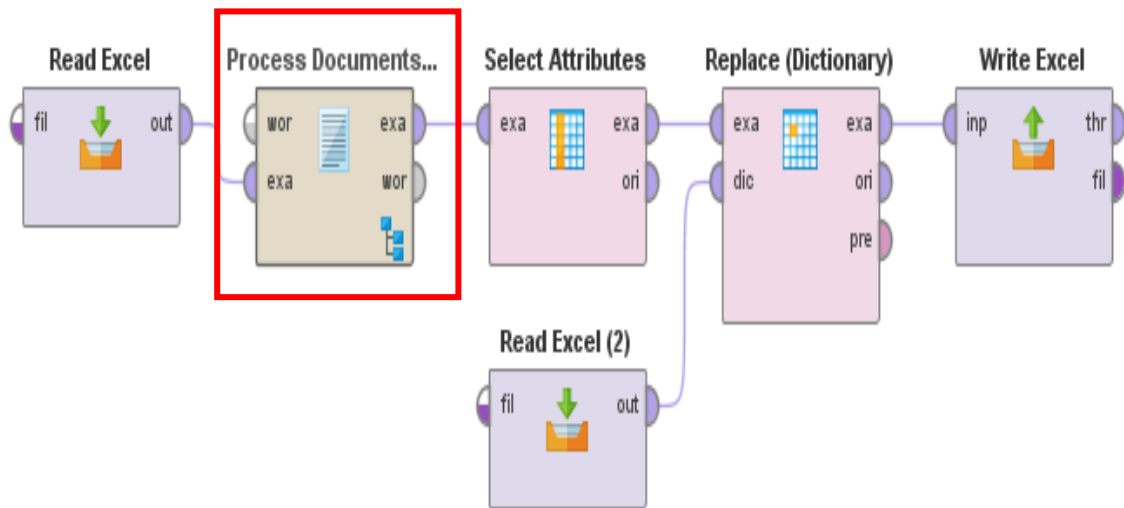
This “read excel” operator retrieved the data from our excel file and imported it to the software.

Row No.	Message
1	m nt goin, got...
2	i dun mind g...
3	goin 4 my driv...
4	Cut nails oso...
5	Den wat will ...
6	m not at hom...
7	m on da bus ...
8	m eatin wif m...
9	Do u noe ho...
10	U meet other ...
11	Dunno i shd ...
12	hv to finish m...
13	I jokin oni
14	Oh... Kay... O...
15	Wat r u up 2 ?
16	lol... ryt thr.
17	u r so sweet..

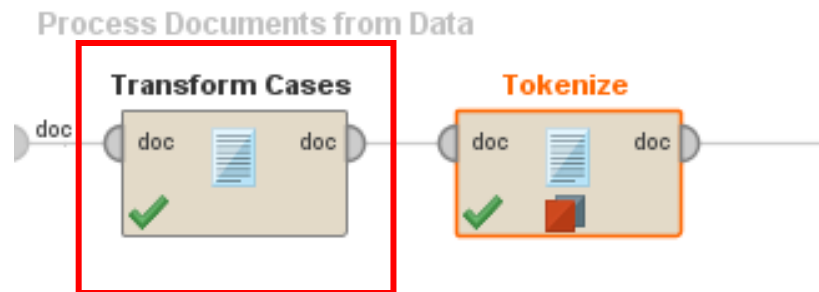
Data is being read by the software.

3. Tokenization:

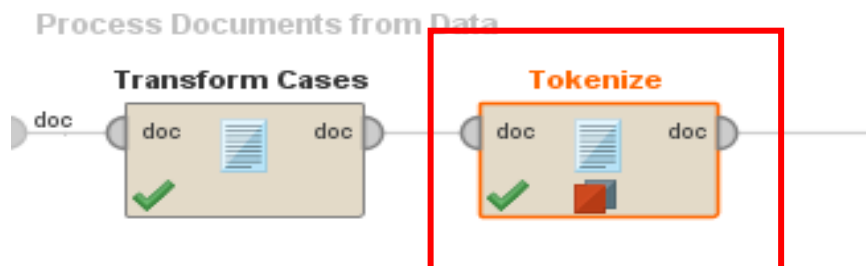
Once we had our finalized data, we firstly transform the cases of each data, as our system is case sensitive and hence, would have caused problem in our future processes. Then, we perform tokenization on the data, converting each of the words in tokens. For this purpose, text processing extension is required on the software.



As tokenization cannot be performed on a dataset, we firstly convert the data into individual Documents using “Process Documents from Data” operator, and inside this operator perform our operation. Here the data is first converted into different documents and then the processes are applied.



Firstly, we transform the cases of our data for easy processing. And then tokenization is performed.



Transformed and Tokenized data:

m nt goin got somethin on

i dun mind goin jazz oso

goin my drivin den go shoppin after tt

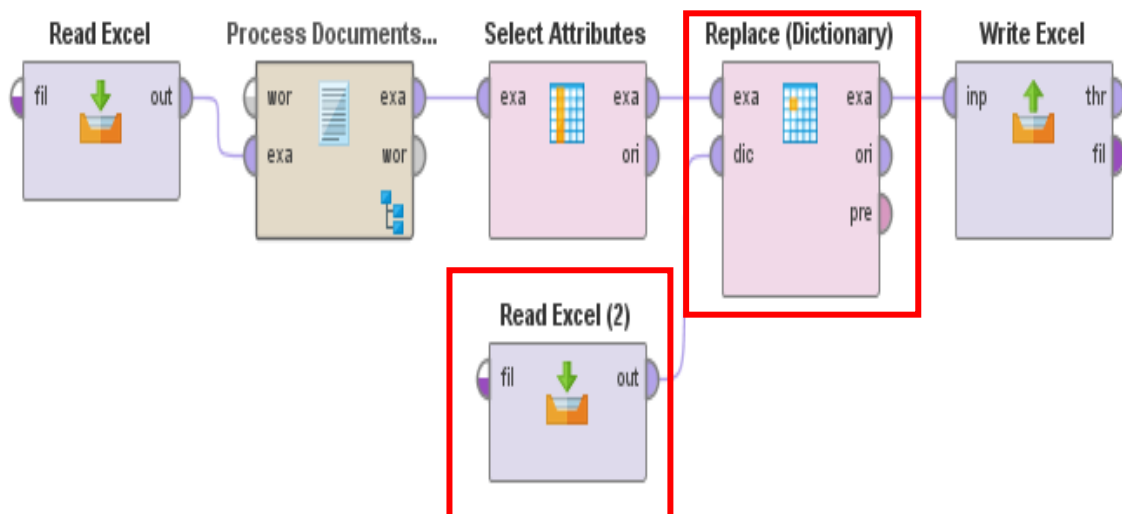
m eatin wif my frens now

m on da bus going home

4. Replacement:

Once the data has been tokenized, we formed a replacing dictionary which consisted of the slangs and their correct words. These tokens were then replaced by the correct word once passed through the dictionary created, giving us the result we needed.

“Read Excel (2)” operator imports the dictionary created, and the “Replace (Dictionary)” operator takes the input of the sms and the dictionary and compares them to give us the corrected data.



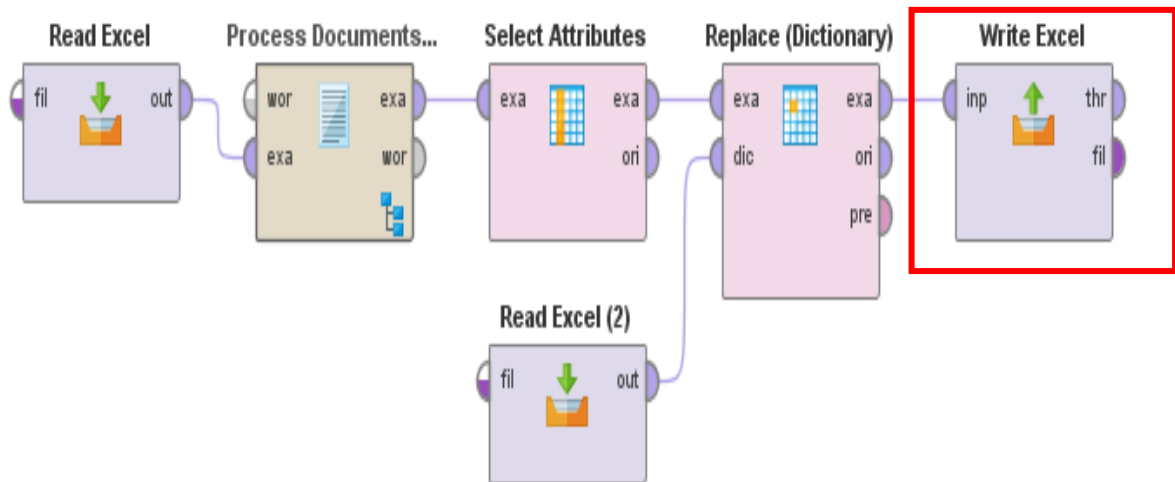
Dictionary:

Ref	Incorrect	Correct
@	\b@\b	at
\bm\b	\bm\b	I'm
0	\b0\b	Zero
1	\b1\b	One
10	\b10\b	Ten
10q	\b10q\b	Thank You
121	\b121\b	One to one
1337	\b1337\b	leet or Elite
1457	\b1457\b	Last
1ce	\b1ce\b	once
2	\b2\b	to
26y4u	\b26y4u\b	too sexy for you
2bono	\b2bono\b	To be or not to be
2day	\b2day\b	today
2mor	\b2mor\b	tomorrow
2n8	\b2n8\b	Two and Eight
2nite	\b2nite\b	tonight
2u2	\b2u2\b	To you too
3	\b3\b	Three
4	\b4\b	for
411	\b411\b	Information
4ever	\b4ever\b	Forever
4frnfr	\b4frnfr\b	Forever and Ever
5	\b5\b	Five
6	\b6\b	Six
7	\b7\b	Seven

This Dictionary consists of about 900+ words and many more can be added according to the different slangs used in different part of the world. Moreover, same word can be written in different forms. Many common words have been listed in this dictionary, having different slangs for the same word.

5. Exporting the data:

After the replacement is performed, the data is then exported to an Excel Workbook using “Write Excel” operator. This enables us to store the output somewhere.

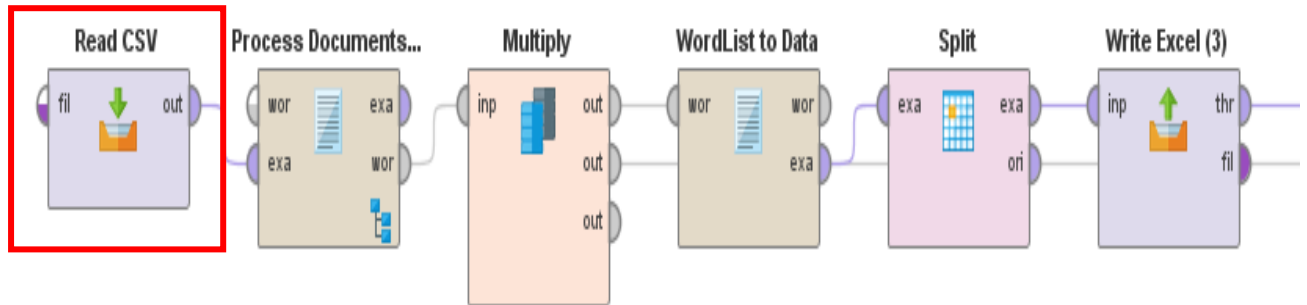


Our output:

text					
I'm not going, got something on					
i don't mind going jazz also.					
going for my driving then go shopping after it					
cut nails also must wait until i finish driving what, lunch still must eat what...					
then what will the schedule be like on sun?					
I'm not at home in the afternoon what.					
I'm on the bus going home .					
I'm eating with my friends now					
do you know how to send files between two computers ?					
you meet other friend don't want meet me ah ... must be a guy right .					
dont know i should be driving or not because i go school One hour only .					
have to finish my film yet ...					

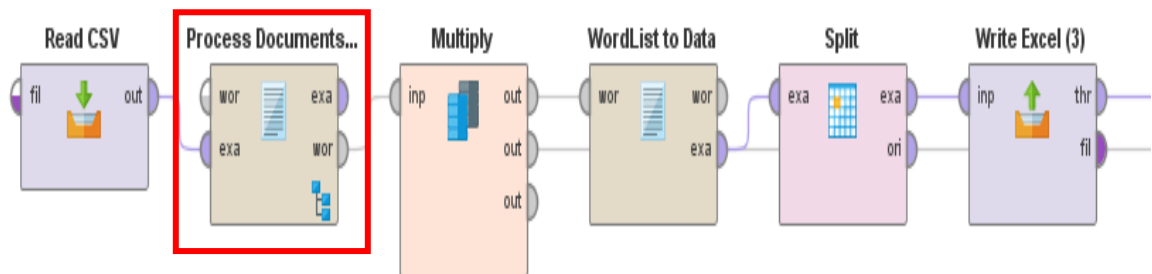
6. Re - Import:

After the standardization is complete, we again import the standardized data, or take another relatively large corpus to perform next word prediction. We firstly convert this corpus into “.csv” format as it makes further processing easier.

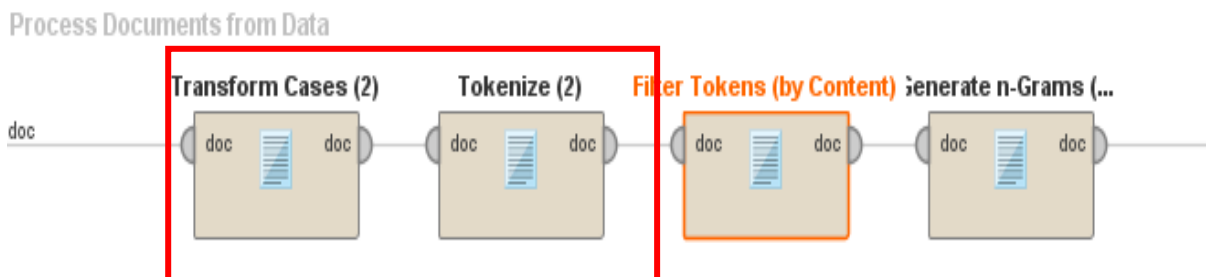


6. Re - Tokenization:

Again, the dataset is first converted into individual documents and then the cases are transformed into lower case and then tokenized.

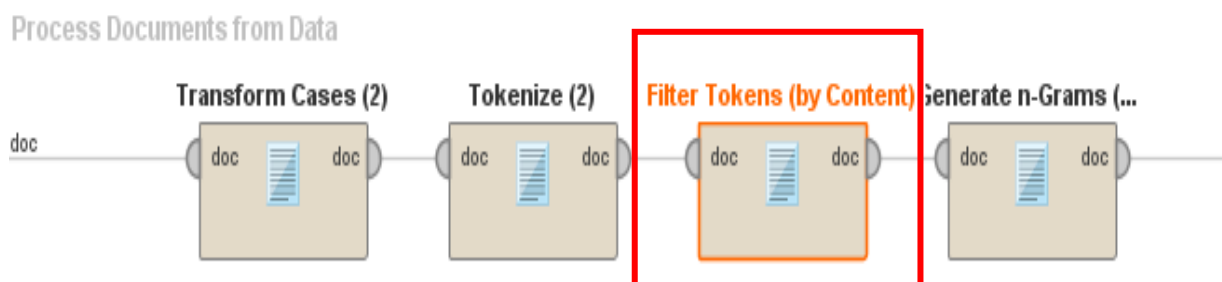


Data converted into Documents and further processed.



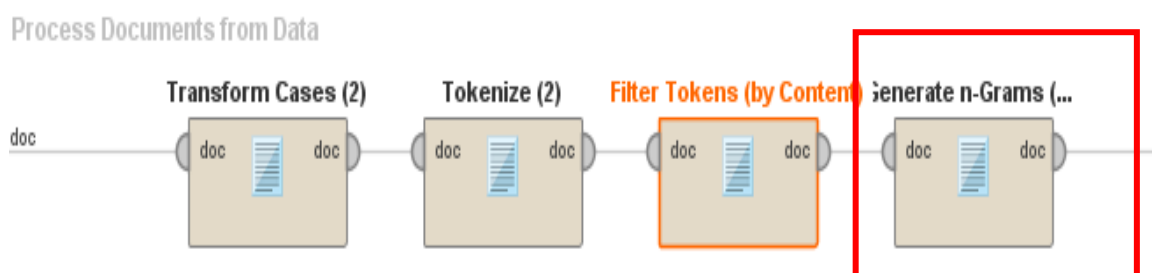
7. Filtering tokens (by Length):

When we perform tokenization, apostrophe etc. are removed, hence making “s” an individual word having no meaning. Hence we remove the single letter words by filtering the tokens by content.



8. Generating n-grams (bigram):

After we have tokenized our data, we generate bigram for our corpus, and also get the frequency of each word and each bigram as well. This will help us determine what word should become the predicted word for our given input.



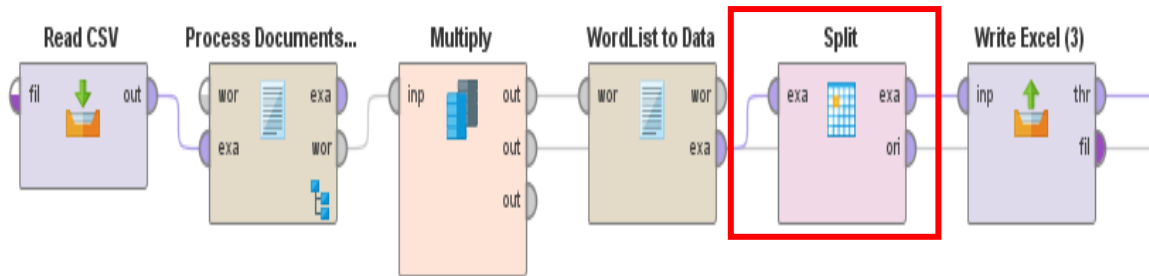
The wordlist created, which consists of total occurrence, and the word is as follows.

Word	Attribute Name	Total Occurences	Document Occurences
a	a	139	96
a_backlash	a_backlash	1	1
a_beehive	a_beehive	1	1
a_big	a_big	1	1
a_bigger	a_bigger	1	1
a_blank	a_blank	1	1
a_block	a_block	1	1
a_bond	a_bond	1	1
a_book	a_book	1	1
a_bottom	a_bottom	1	1
a_broad	a_broad	1	1
a_buzz	a_buzz	1	1
a_cardinal	a_cardinal	1	1
a_career	a_career	1	1
a_category	a_category	1	1

Here, as we can see, various words in our documents, corresponding to “a” are displayed along with the occurring frequency.

9. Splitting the bigrams:

For reading this data on database and easily retrieving the required value, we split our bigrams in two separate columns using the “Split” operator.



Now our output looks like:

in documents	total	word_1	word_2
96	139	a	?
1	1	a	backlash
1	1	a	beehive
1	1	a	big
1	1	a	bigger
1	1	a	blank
1	1	a	block
1	1	a	bond
1	1	a	book
1	1	a	bottom
1	1	a	broad
1	1	a	buzz

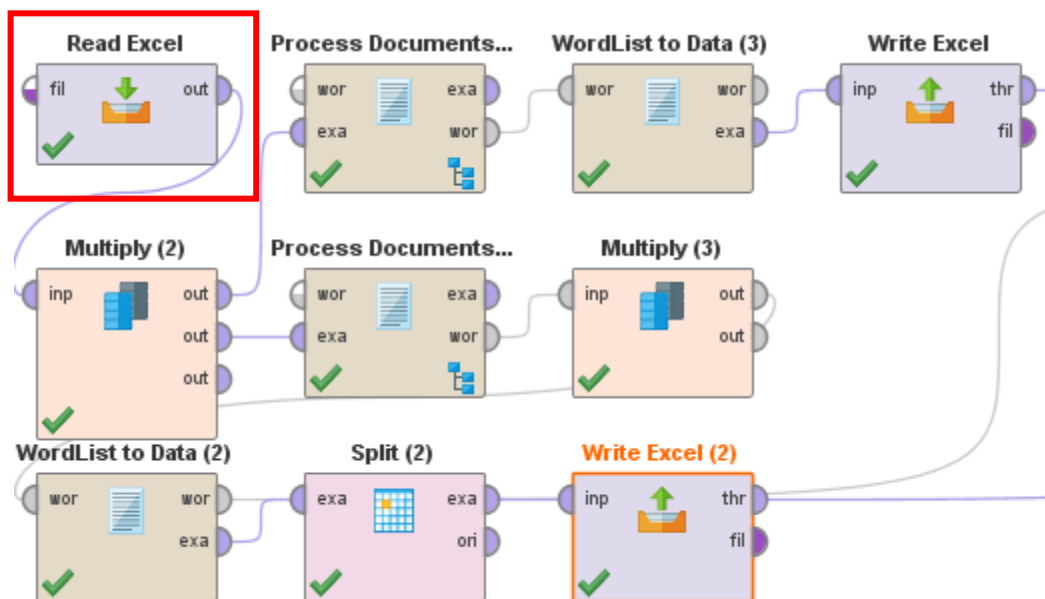
10. Loading on database:

Using XAMPP MySQL server, this data was loaded on a database.

→T←			documents	total	word_1	word_2
<input type="checkbox"/>			96	139	a	NULL
<input type="checkbox"/>			1	1	a	backlash
<input type="checkbox"/>			1	1	a	beehive
<input type="checkbox"/>			1	1	a	big
<input type="checkbox"/>			1	1	a	bigger
<input type="checkbox"/>			1	1	a	blank
<input type="checkbox"/>			1	1	a	block
<input type="checkbox"/>			1	1	a	bond
<input type="checkbox"/>			1	1	a	book
<input type="checkbox"/>			1	1	a	bottom
<input type="checkbox"/>			1	1	a	broad
<input type="checkbox"/>			1	1	a	buzz
<input type="checkbox"/>			1	1	a	cardinal
<input type="checkbox"/>			1	1	a	career

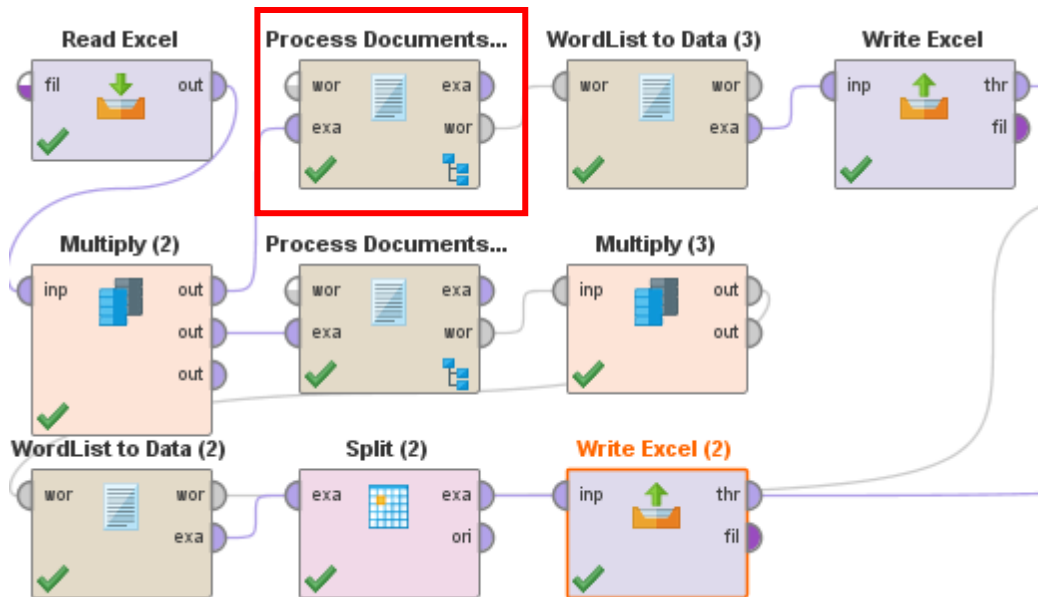
11. Importing corrected data:

In order to get higher efficiency, we use the next word prediction to fix error. In order to do that, we import this data using “Read Excel” operator.

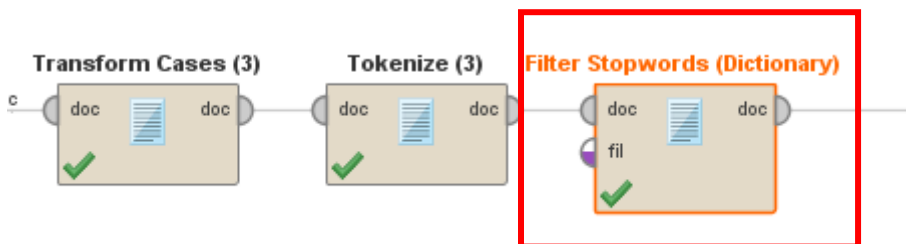


12. Filtering out English words:

This data is again converted into individual documents, tokenized and then all words existing in an English dictionary are filtered out, leaving only the words with error in it.



Using “Filter Stopwords (dictionary)” we import an English dictionary in it, hence filtering out all English words.

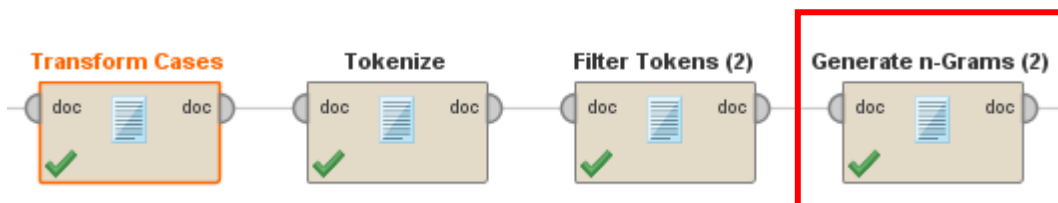
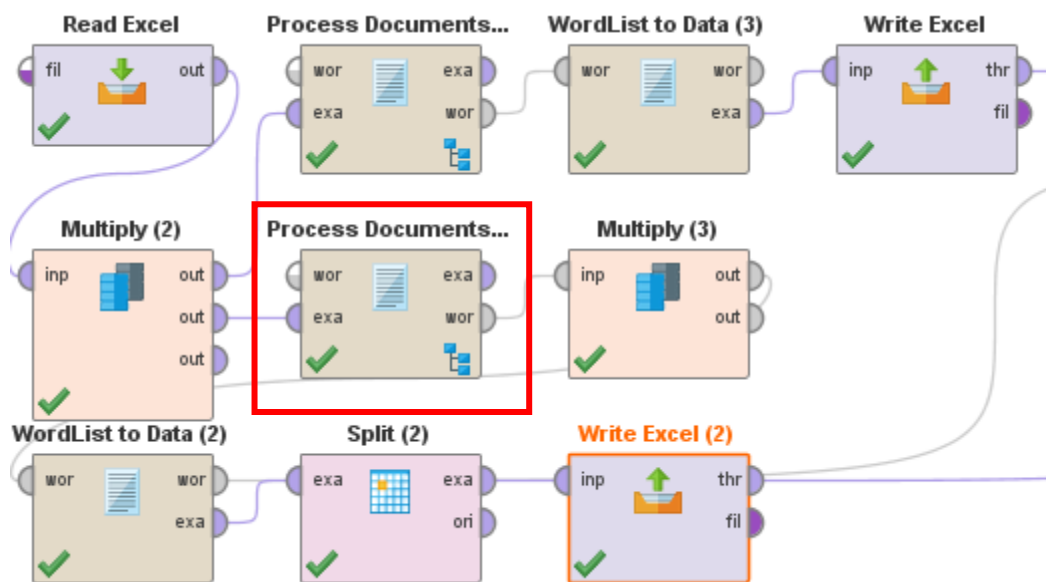


This process extracts all non corrected words. Once these words are identified, these can be used to detect correct word from our “prediction” database.

word	in docume	total
anythin	1.0	1.0
canit	2.0	2.0
mys	1.0	1.0
thanx	1.0	1.0

13. Generating n-grams on corrected data:

We process the same data and tokenize it. After tokenization we perform bigram on it so that we could create another database and retrieve the previous word of uncorrected data with the help of identified errors.



We get the final output after performing “split” operation.

in document	total	word_1	word_2
3.0	3.0	a	
1.0	1.0	a	guy
1.0	1.0	a	reason
1.0	1.0	a	week
1.0	1.0	about	
1.0	1.0	about	it
2.0	2.0	after	
1.0	1.0	after	a
1.0	1.0	after	it
2.0	2.0	afternoon	
1.0	1.0	afternoon	message
1.0	1.0	afternoon	what
1.0	1.0	ago	
1.0	1.0	ago	already
1.0	1.0	ah	
1.0	1.0	ah	must
4.0	4.0	all	
2.0	2.0	all	correct
1.0	1.0	all	the
1.0	1.0	all	too
1.0	1.0	almost	
1.0	1.0	almost	done

14. Predicting the correct word:

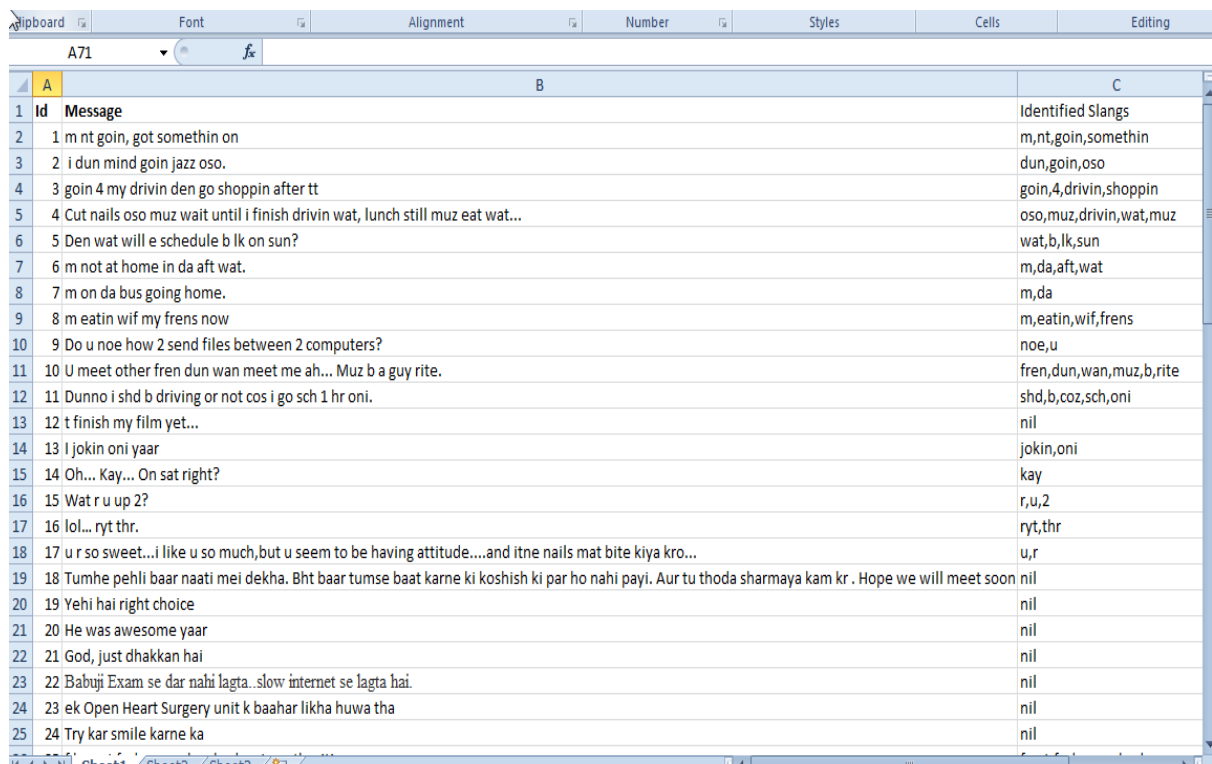
Now once we have the knowledge which words are not corrected, we can retrieve those words and check them in the second database to find out their preceding words. Once we have the previous word, we look into our “prediction” database and find the words corresponding to it. Once we have the list of possible words, the user can choose the word.

4. PERFORMANCE ANALYSIS

We sent the database to 3 users, who filled up the form identifying the slangs used in each data line and writing their corresponding meanings according to their knowledge. These slangs with their corresponding meanings were tested against the database for slangs made by us to automatically convert the slang into formal language. The accuracy of the database is thus tested and can be now imported into the Rapid Miner Tool to filter the data into correct language.

4.1 ANALYSIS

USER1:



Id	Message	Identified Slangs
1	m nt goin, got somethin on	m,nt,goin,somethin
2	i dun mind goin jazz oso.	dun,goin,oso
3	goin 4 my drivin den go shoppin after tt	goin,4,drivin,shoppin
4	Cut nails oso muz wait until i finish drivin wat, lunch still muz eat wat...	oso,muz,drivin,wat,muz
5	Den wat will e schedule b lk on sun?	wat,b,lk,sun
6	m not at home in da aft wat.	m,da,aft,wat
7	m on da bus going home.	m,da
8	m eatin wif my frens now	m,eatin,wif,frens
9	Do u noe how 2 send files between 2 computers?	noe,u
10	U meet other fren dun wan meet me ah... Muz b a guy rite.	fren,dun,wan,muz,b,rite
11	Dunno i shd b driving or not cos i go sch 1 hr oni.	shd,b,coz,sch,oni
12	t finish my film yet...	nil
13	I jokin oni yaar	jokin,oni
14	Oh... Kay... On sat right?	kay
15	Wat r u up 2?	r,u,2
16	lol... ryt thr.	ryt,thr
17	u r so sweet...i like u so much,but u seem to be having attitude....and itne nails mat bite kiya kro...	u,r
18	Tumhe pehli baar naati mei dekha. Bht baar tumse baat karne ki koshish ki par ho nahi payi. Aur tu thoda sharmaya kam kr . Hope we will meet soon	nil
19	Yehi hai right choice	nil
20	He was awesome yaar	nil
21	God, just dhakkan hai	nil
22	Babuji Exam se dar nahi lagta .slow internet se lagta hai.	nil
23	ek Open Heart Surgery unit k baahar likha huwa tha	nil
24	Try kar smile karne ka	nil

Data and its slangs recognized

	C	D	E	F	G	H
1	Identified Slangs	Corresponding meanings				
2	m,nt,goin,somethin	am,not,going,something				
3	dun,goin,oso	don't,going,also				
4	goin,4,drivin,shoppin	going,for,driving,shopping				
5	oso,muz,drivin,wat,muz	also,must,driving,what,must				
6	wat,b,lk,sun	what,be,like,Sunday				
7	m,da,aft,wat	am,the,after,what				
8	m,da	am,the				
9	m,eatin,wif,frens	am,eating,with,friends				
10	noe,u	know,you				
11	fren,dun,wan,muz,b,rite	friends,don't,want, must, be, right				
12	shd,b,coz,sch,oni	should,be,because,school,only				
13	nil	nil				
14	jokin,oni	joking,only				
15	kay	OK				
16	r,u,2	are,you,too				
17	ryt,thr	right, there				
18	u,r	you,are				
19	nil	nil				
20	nil	nil				
21	nil	nil				
22	nil	nil				
23	nil	nil				
24	nil	nil				
25	nil	nil				

Slangs along with their corresponding meanings

	A	B	C
26	25	f he cnt fnd a resn, he shud go to authorities	f,cnt,fnd,resn,shud
27	26	Ard 6 like dat	ard
28	27	But i hv enuff space got like 4 mb	hv,enuff
29	28	Where r e meeting tmr?	r,e,tmr
30	29	Il b going 2 sch on fri quite early lor cos mys sis got paper in da morn	ll,b,sch,fri,cos,sis,da,morn
31	30	I'll ttyl, bzy ryt nw	ttyl,bzy,ryt,nw
32	31	wat u wana do?	wat,u,wana
33	32	All e best 4 ur driving tmr	e,ur,tmr
34	33	Ok but tell me half an hr b4 u come i need 2 prepare	b4,u,2
35	34	Yar... I tot u knew dis would happen long ago already.	tot,u,dis
36	35	Ok... Take ur time n enjoy ur dinner...	ur,n
37	36	Yup... We r at 2nd floor boots...	r,yup
38	37	u reach home den let me noe	u,den,noe
39	38	U gt to tl dem its nt ryt.	gt,tl,nt,ryt
40	39	stp laughin at me.	stp,laughin
41	40	hy, sry cnt meet u today.	hy,sry,cnt,u
42	41	im almst dne.	im,almst,dne
43	42	Nope din buy anythin.	din,anythin
44	43	muz b still enjoyin herself	muz,b,enjoyin
45	44	whr hv u been?	whr,hv,u
46	45	My sis will drive me dere	sis,dere
47	46	I oso wont stay 4 too long	oso,4
48	47	Wat time is ur tuition?	wat,ur
49	48	V to meet em at 845	v,em
50	49	m comin bk nw.	m,comin,bk,nw

	C	D	E	F	G	H
26	f,cnt,fnd,rsn,shud	if, can't,find,reason,should				
27	ard	and				
28	hv,enuff	have,enough				
29	r,e,tmr	are,the,tomorrow				
30	ll,b,sch,fri,cos,sis,da,morn	I'll,be,school,Friday,because,sister,the,morning				
31	ttyl,bzy,ryt,nw	talk to you later,busy,right,now				
32	wat,u,wana	what,you,want to				
33	e,ur,tmr	the,your,tomorrow				
34	b4,u,2	before,you,too				
35	tot,u,dis	thought,you,this				
36	ur,n	you're,and				
37	r,yup	are,yes				
38	u,den,noe	you,the,know				
39	gt,tl,nt,ryt	got,tell,not,right				
40	stp,laughin	stop,laughing				
41	hy,sry,cnt,u	hey,sorry,can't,you				
42	im,almst,dne	I'm,almost,done				
43	din,anythin	did not,anything				
44	muz,b,enjoyin	must, be , enjoying				
45	whr,hv,u	where,have,you				
46	sis,dere	sister,there				
47	oso,4	also, for				
48	wat,ur	what, your				
49	v,em	we, them				
50	m,comin,bk,nw	I'm,coming,back,now				

	A	B	C
51	50 No biggy.		nil
52	51 Holidays included hai?		nil
53	52 Afr a week il cm hme.		il,cm,hme
54	53 mom is n hr way.		n,hr
55	54 Prepared exams ke liye?		nil
56	55 Intruder dekh le		nil
57	56 m comin soon, gota wait 4 my sis first		m,comin,gota,4,sis
58	57 m in d car ... Cant send u.		m,d,u
59	58 Thanx 4 d gift...Muz tell me when u free		thanx,4,d,muz
60	59 ll present nxt mon. Den was wonderin how come not on fri		ll,nxt,mon,den,wonderin,fri
61	60 Got pts one huh... U joinin, if u r i dun mind		pts,u,joinin,r,dun
62	61 ll mt u all too.		ll,mt,u
63	62 idk wat u sayin.		idk,wat,u,sayin
64	63 brb. I gtg.		brb,gtg
65	64 s nt angry w his bf since he came to pick her yest.		nt,bf,yest
66	65 ll msg u to kp u informed		ll,msg,u,kp
67	66 But u muz tell me wat u wan to noe		u,muz,wat,wan,noe
68	67 Hope u find wot was lost		u,wot
69	68 s yr hols been girl		nil
70	69 did u c my aftn msg?		u,c,aftn,msg
71			
72			
73			
74			

A71		
	C	D
51	nil	nil
52	nil	nil
53	il,cm,hme	I'll,come,home
54	n,hr	an, hour
55	nil	nil
56	nil	nil
57	m,comin,gota,4,sis	I'm,coming,gotto,for,sister
58	m,d,u	am,the,you
59	thanx,4,d,muz	thanks,for,the,must
60	ll,nxt,mon,den,wonderin,fri	I'll,next,Monday,then,wondering,Friday
61	pts,u,joinin,r,dun	pts,you,joining,are,done
62	ll,mt,u	I'll,meet,you
63	idk,wat,u,sayin	I don't know,what,you,saying
64	brb,gtg	be right back,got to go
65	nt,bf,yest	not,boyfriend,yesterday
66	ll,msg,u,kp	I'll, message, you, keep
67	u,muz,wat,wan,noe	you,must,what,want to, know
68	u,wot	you,what
69	nil	nil
70	u,c,aftn,msg	you,see,afternoon,message
71		
72		
73		
74		

USER2:

	A	B	C
26	25 f he cnt fnd a resn, he shud go to authorities		f,cnt,fnd,resn,shud
27	26 Ard 6 like dat		ard
28	27 But i hv enuff space got like 4 mb		hv,enuff
29	28 Where r e meeting tmr?		r,e,tmr
30	29 Il b going 2 sch on fri quite early lor cos mys sis got paper in da morn		ll,b,sch,fri,cos,sis,da,morn
31	30 I'll ttyl, bzy ryt nw		ttyl,bzy,ryt,nw
32	31 wat u wana do?		wat,u,wana
33	32 All e best 4 ur driving tmr		e,ur,tmr
34	33 Ok but tell me half an hr b4 u come i need 2 prepare		b4,u,2,hr
35	34 Yar... I tot u knew dis would happen long ago already.		tot,u,dis
36	35 Ok... Take ur time n enjoy ur dinner...		ur,n
37	36 Yup... We r at 2nd floor boots...		r,yup
38	37 u reach home den let me noe		u,den,noe
39	38 U gt to tl dem its nt ryt.		gt,tl,nt,ryt
40	39 stp laughin at me.		stp,laughin
41	40 hy, sry cnt meet u today.		hy,sry,cnt,u
42	41 im almst dne.		im,almst,dne
43	42 Nope din buy anythin.		din,anythin
44	43 muz b still enjoyin herself		muz,b,enjoyin
45	44 whr hv u been?		whr,hv,u
46	45 My sis will drive me dere		sis,dere
47	46 I oso wont stay 4 too long		oso,4
48	47 Wat time is ur tuition?		wat,ur
49	48 V to meet em at 845		v,em
50	49 m comin bk nw.		m,comin,bk,nw

	A	B	C
1	Id	Message	Identified Slangs
2	1	m nt goin, got somethin on	m,nt,goin,somethin
3	2	i dun mind goin jazz oso.	dun,goin,oso
4	3	goin 4 my drivin den go shoppin after tt	goin,4,drivin,shoppin
5	4	Cut nails oso muz wait until i finish drivin wat, lunch still muz eat wat...	oso,muz,drivin,wat,muz
6	5	Den wat will e schedule b lk on sun?	wat,b,lk,sun
7	6	m not at home in da aft wat.	m,da,aft,wat
8	7	m on da bus going home.	m,da
9	8	m eatin wif my frens now	m,eatin,wif,frens
10	9	Do u noe how 2 send files between 2 computers?	noe,u
11	10	U meet other fren dun wan meet me ah... Muz b a guy rite.	fren,dun,wan,muz,b,rite
12	11	Dunno i shd b driving or not cos i go sch 1 hr oni.	shd,b,coz,sch,oni
13	12	t finish my film yet...	nil
14	13	I jokin oni yaar	jokin,oni
15	14	Oh... Kay... On sat right?	kay
16	15	Wat r u up 2?	r,u,2
17	16	lol... ryt thr.	ryt,thr,lol
18	17	u r so sweet...i like u so much,but u seem to be having attitude....and itne nails mat bite kiya kro...	u,r
19	18	Tumhe pehli baar naati mei dekha. Bht baar tumse baat karne ki koshish ki par ho nahi payi. Aur tu thoda sharmaya kam kr . Hope we will meet soon	nil
20	19	Yehi hai right choice	nil
21	20	He was awesome yaar	nil
22	21	God, just dhakkan hai	nil
23	22	Babuji Exam se dar nahi lagta..slow internet se lagta hai.	nil
24	23	ek Open Heart Surgery unit k baahar likha huwa tha	nil

User2.xlsx - Microsoft Excel Starter

A	B	C
51	50 No biggy.	nil
52	51 Holidays included hai?	nil
53	52 Afr a week il cm hme.	il,cm,hme,aftr
54	53 mom is n hr way.	n,hr
55	54 Prepared exams ke liye?	nil
56	55 Intruder dekh le	nil
57	56 m comin soon, gota wait 4 my sis first	m,comin,gota,4,sis
58	57 m in d car ... Cant send u.	m,d,u
59	58 Thanx 4 d gift...Muz tell me when u free	thanx,4,d,muz
60	59 Il present nxt mon. Den was wonderin how come not on fri	ll,nxt,mon,den,wonderin,fr
61	60 Got pts one huh... U joinin, if u r i dun mind	pts,u,joinin,r,dun
62	61 ll mt u all too.	ll,mt,u
63	62 idk wat u sayin.	idk,wat,u,sayin
64	63 brb. I gtg.	brb,gtg
65	64 s nt angry w his bf since he came to pick her yest.	nt,bf,yest
66	65 ll msg u to kp u informed	ll,msg,u,kp
67	66 But u muz tell me wat u wan to noe	u,muz,wat,wan,noe
68	67 Hope u find wot was lost	u,wot
69	68 s yr hols been girl	nil
70	69 did u c my aftn msg?	u,c,aftn,msg
71		
72		
73		
74		
75		

B75

	C	D
1	Identified Slangs	Corresponding meanings
2	m,nt,goin,somethin	am,not,going,something
3	dun,goin,oso	don't,going,also
4	goin,4,drivin,shoppin	going,for,driving,shopping
5	oso,muz,drivin,wat,muz	also,must,driving,what,must
6	wat,b,lk,sun	what,be,like,Sunday
7	m,da,aft,wat	am,the,after,what
8	m,da	am,the
9	m,eatin,wif,frens	am,eating,with,friends
10	noe,u	know,you
11	fren,dun,wan,muz,b,rite	friends,don't,want, must, be , right
12	shd,b,coz,sch,oni	should,be,because,school,only
13	nil	nil
14	jokin,oni	joking,only
15	kay	OK
16	r,u,2	are,you,too
17	ryt,thr,lol	right, there,laugh out loud
18	u,r	you,are
19	nil	nil
20	nil	nil
21	nil	nil
22	nil	nil
23	nil	nil
24	nil	nil
25	nil	nil

	C	D
26	f,cnt,fnd,resn,shud	if, can't,find,reason,should
27	ard	and
28	hv,enuff	have,enough
29	r,e,tmr	are,we,tomorrow
30	ll,b,sch,fri,cos,sis,da,morn	I'll,be,school,Friday,because,sister,the,morning
31	ttyl,bzy,ryt,nw	talk to you later,busy,right,now
32	wat,u,wana	what,you,want to
33	e,ur,tmr	the,your,tomorrow
34	b4,u,2,hr	before,you,too,hour
35	tot,u,dis	thought,you,this
36	ur,n	your,and
37	r,yup	are,yes
38	u,den,noe	you,the,know
39	gt,tl,nt,ryt	got,tell,not,right
40	stp,laughin	stop,laughing
41	hy,sry,cnt,u	hey,sorry,can't,you
42	im,almst,dne	I'm,almost,done
43	din,anythin	did not,anything
44	muz,b,enjoyin	must, be , enjoying
45	whr,hv,u	where,have,you
46	sis,dere	sister,there
47	oso,4	also, for
48	wat,ur	what, your
49	v,em	we, them
50	m,comin,bk,nw	I'm,coming,back,now
51	nil	nil
52	nil	nil
53	il,cm,hme,aftr	I'll,come,home,after
54	n,hr	an, hour
55	nil	nil
56	nil	nil
57	m,comin,gota,4,sis	I'm,coming,gotto,for,sister
58	m,d,u	am,the,you
59	thanx,4,d,muz	thanks,for,the,must
60	ll,nxt,mon,den,wonderin,fri	I'll,next,Monday,then,wondering,Friday
61	pts,u,joinin,r,dun	pts,you,joining,are,done
62	ll,mt,u	I'll,meet,you
63	idk,wat,u,sayin	I don't know,what,you,saying
64	brb,gtg	be right back,got to go
65	nt,bf,yest	not,boyfriend,yesterday
66	ll,msg,u,kp	I'll, message, you, keep
67	u,muz,wat,wan,noe	you,must,what,want to, know
68	u,wot	you,what
69	nil	nil
70	u,c,aftn,msg	you,see,afternoon,message

USER3:

Protected View This file originated from an Internet location and might be unsafe. Click for more details. Enable Editing		
D71 haven't,thought,about		
A	B	C
1	ld Message	Slangs
2	1 m nt goin, got somethin on	m,nt,goin,somethin
3	2 i dun mind goin jazz oso.	dun, goin, oso
4	3 goin 4 my drivin den go shoppin after tt	goin, 4, drivin, den, shoppin
5	4 Cut nails oso muz wait until i finish drivin wat, lunch still muz eat wat...	oso, muz, drivin, wat
6	5 Den wat will e schedule b lk on sun?	den,wat,e,b,lk,sun
7	6 m not at home in da aft wat.	m,da,aft, wat
8	7 m on da bus going home.	m,da
9	8 m eatin wif my frens now	m,eatin,wif,frens
10	9 Do u noe how 2 send files between 2 computers?	u,noe,2,
11	10 U meet other fren dun wan meet me ah... Muz b a guy rite.	u,fren,dun,wan,muz,b,rite
12	11 Dunno i shd b driving or not cos i go sch 1 hr oni.	dunno, shd,b,cos,sch,hr,on
13	12 t finish my film yet...	t
14	13 I jokin oni yaar	jokin,oni
15	14 Oh... Kay... On sat right?	kay, sat
16	15 Wat r u up 2?	wat,r,u.2
17	16 lol... ryt thr.	lol,ryt,thr
18	17 u r so sweet...i like u so much,but u seem to be having attitude....and itne nails mat bite kiya kro...	u,r
19	18 Tumhe pehli baar naati mei dekha. Bht baar tumse baat karne ki koshish ki par ho nahi payi. Aur tu thoda sharmaya kam kr . Hope we will meet soon.	(-)
20	19 Yehi hai right choice	(-)
21	20 He was awesome yaar	(-)
22	21 God, just dhakkan hai	(-)
23	22 Babuji Exam se dar nahi lagta..slow internet se lagta hai.	(-)

C33		fx e,4,ur,tmr
A	B	C
22	Babuji Exam se dar nahi lagta .slow internet se lagta hai.	(-)
23	ek Open Heart Surgery unit k baahar likha huwa tha	(-)
24	Try kar smile karne ka	(-)
25	f he cnt fnd a resn, he shud go to authorities	f,cnt,fnd,resn,shud
26	Ard 6 like dat	ard
27	But i hv enuff space got like 4 mb	hv,enuff
28	Where r e meeting tmr?	r,e,tmr
29	Il b going 2 sch on fri quite early lor cos mys sis got paper in da morn	il,b,2,sch,fri,cos,mys,sis,da
30	I'll ttyl, bzy ryt nw	ttyl,bzy,ryt,nw
31	wat u wana do?	wat,u,wana
32	All e best 4 ur driving tmr	e,4,ur,tmr
33	Ok but tell me half an hr b4 u come i need 2 prepare	hr,b4,u,2
34	Yar... I tot u knew dis would happen long ago already.	tot,u,dis
35	Ok... Take ur time n enjoy ur dinner...	ur,n
36	Yup... We r at 2nd floor boots...	r
37	u reach home den let me noe	u,den,noe
38	U gt to tl dem its nt ryt.	u,gt,tl,dem,nt,ryt
39	stp laughin at me.	stp,laughin
40	hy, sry cnt meet u today.	hy,sry,cnt,u
41	im almst dne.	lm,almst,dne
42	Nope din buy anythin.	din,anythin
43	muz b still enjoyin herself	mus,b,enjoyin

Protected View This file originated from an Internet location and might be unsafe. Click for more details.

C33		fx e,4,ur,tmr
A	B	C
43	muz b still enjoyin herself	mus,b,enjoyin
44	whr hv u been?	whr,hv,i
45	My sis will drive me dere	sis,dere
46	I oso wont stay 4 too long	oso,4
47	Wat time is ur tuition?	wat,ur
48	V to meet em at 845	v,em
49	49 m comin bk nw.	m,comin,bk,nw
50	No biggy.	biggy
51	Holidays included hai?	(-)
52	Aftr a week il cm hme.	aftr,il,cm,hme
53	53 mom is n hr way.	n,hr
54	Prepared exams ke liye?	(-)
55	Intruder dekh le	(-)
56	56 m comin soon, gota wait 4 my sis first	m,comin,gota,4,sis
57	57 m in d car ... Cant send u.	m,d,u
58	58 Thanx 4 d gift...Muz tell me when u free	thanx,4,d,muz,u
59	59 Il present nxt mon. Den was wonderin how come not on fri	il,nxt,mon,den,wonderin,fi
60	60 Got pts one huh... U joinin, if u r i dun mind	u,r,dun
61	61 Il mt u all too.	il,mt,u
62	62 idk wat u sayin.	idk,wat,u,sayin
63	63 brb. I gtg.	brb,gtg
64	64 s nt angry w her bf since he came to pick her yest.	s,nt,w,bf,yest
65	65 Il msg u to kp u informed	il,msg,u,kp
66	66 But u muz tell me wat u wan to noe	u,muz,wat,wan,noe
67	67 Hope u find wot was lost	u,wot
68	68 s yr hols been girl	s,yr,hols
69	69 did u c my aftn msg?	u,c,aftn,msg
70	70 + tot abt it	+ tot abt

Protected View This file originated from an Internet location and might be unsafe. Click for more details. Enable Editing

	C	D	E	F	G	H	I	J	K	L
1	Slangs	Correct								
2	m,nt,goin,somethin	I'm, not, going, something								
3	dun, goin, oso	don't, going, also								
4	goin, 4, drivin, den, shoppin	going, for, driving, then, shopping								
5	oso, muz, drivin, wat	also, must, driving, what								
6	den,wat,e,b,lk,sun	then,what,the,be,like,Sunday								
7	m,da,aft, wat	I'm, the,afternoon,what								
8	m,da	I'm, the								
9	m,eatin,wif,frens	I'm,eating,with,friends								
10	u,noe,2,	you,know,to								
11	u,fren,dun,wan,muz,b,rite	you,friend,don't,want,must,be,right								
12	dunno, shd,b,cos,sch,hr,on	don't know, should,be,because,school,hour,only								
13	t	not								
14	jokin,oni	joking,only								
15	kay, sat	okay, Saturday								
16	wat,r,u,2	what,are,you,to								
17	lol,ryt,thr	laugh out loud, right, there								
18	u,r	you,are								
19	(-)	(-)								
20	(-)	(-)								
21	(-)	(-)								
22	(-)	(-)								
23	(-)	(-)								
24	(-)	(-)								
25	(-)	(-)								
26	f,cnt,fnd,resn,shud	if,cant,find,reason,should								
27	ard	around								
28	hv,enuff	have,enough								

Protected View This file originated from an Internet location and might be unsafe. Click for more details. Enable Editing

	C	D	E	F	G	H	I	J
29	r,e,tmr	are,we,tomorrow						
30	il,b,2,sch,fri,cos,mys,sis,da,	I'll, be,to,school,because,my,sister,the,morning						
31	ttyl,bzy,ryt,nw	talk,to,you,later,busy,right,now						
32	wat,u,wana	what,you,want to						
33	e,4,ur,tmr	the,for,your,tomorrow						
34	hr,b4,u,2	hour,before,you,to						
35	tot,u,dis	thought,you,this						
36	ur,n	your,and						
37	r	are						
38	u,den,noe	you,then,know						
39	u,gt,tl,dem,nt,ryt	you,got,tell,them,not,right						
40	stp,laughin	stop,laughing						
41	hy,sry,cnt,u	hey,sorry,cant,you						
42	lm,almst,dne	I'm,almost,done						
43	din,anythin	didn't,anything						
44	mus,b,enjoyin	must,be,enjoying						
45	whr,hv,i	where,have,you						
46	sis,dere	sister,there						
47	oso,4	also,for						
48	wat,ur	what,your						
49	v,em	have,them						
50	m,comin,bk,nw	I'm,coming,back,now						
51	biggy	bid deal						
52	(-)							
53	aftr,il,cm,hme	after,i'll,come, home						
54	n,hr	on,her						

49	v,em	have,them		
50	m,comin,bk,nw	I'm,coming,back,now		
51	biggy	bid deal		
52	(-)			
53	aftr,il,cm,hme	after,i'll,come, home		
54	n,hr	on,her		
55	(-)			
56	(-)			
57	m,comin,gota,4,sis	I'm,coming,got,for,sister		
58	m,d,u	I'm,the,you		
59	thanx,4,d,muz,u	thank you,for,the,must,you		
60	il,nxt,mon,den,wonderin,fi	I'll,next,Monday,then,wondering,Friday		
61	u,r,dun	you,are,don't		
62	il,mt,u	I'll,meet,you		
63	idk,wat,u,sayin	I don't know,what,you,saying		
64	brb,gtg	be right back,got to go		
65	s,nt,w,bf,yest	she's,not,with,boyfriend,yesterday		
66	il,msg,u,kp	I'll,message,you,keep		
67	u,muz,wat,wan,noe	you,must,what,want,now		
68	u,wot	you,what		
69	s,yr,hol	how's,your,holidays		
70	u,c,aftn,msg	you,see,afternoon,message		
71	t,tot,abt	haven't,thought,about		

Data Dictionary:

Our dictionary consists of about 1000 words, which include popular slangs.

Few of them being:

Gnite: goodnight

Dnt: don't

Wat: what

Aftn: afternoon

Wot: what

Mon: Monday

Y: why

Fri: Friday

Whr: where

Nyt: night

Wen: when

Lst: last

4.2 ACCURACY:

Accuracy refers to the closeness of a measured value to a standard or known value. In our case we compare the results in our forms to the self-made dictionary for the slangs that are usually used on an everyday basis.

The formula we will use to calculate it:

$$\text{Accuracy} = (\text{Total no. of values} - \text{No. of errors}) * 100 / \text{total no. of values}$$

After measuring each user's answers against the database:

Total no. of errors including USER1, USER2 & USER3= 124

Total no. of values=124

$$\text{Accuracy} = (124-4) * 100 / 124 \Rightarrow 94.4\%$$

Note: Because of survey of a limited dataset, the accuracy has come out to be 94.4%, this is so because most common slangs for a particular area is used for common purposes. The remaining discrepancy is due to different perceptions for different slangs which is where the main challenge for normalization lies.

true positive (TP): equiv. . with hit

true negative (TN): equiv. with correct rejection

false positive (FP): equiv. with false alarm

false negative (FN): equiv. with miss,

PRECISION:

$$t_p / (t_p + f_p) = 122 / (122+2) = 98.38\%$$

RECALL :

$$t_p / (t_p + f_n) = 122 / (122+1) = 99.18\%$$

5. CONCLUSION

In this Project we have learnt the Rapid Miner tool, read research papers on NLP and studied about the work across the world on various language processing techniques. We have also searched and collected the required dataset of multiple text messages; filtered it in order to concentrate the content down. We have then imported it and tokenized it using Rapid Miner tool, finally creating slang to formal language dictionary and testing its accuracy by comparing various responses of users.

After this first phase was over, we took a proper English dataset, used n-grams on it so as to detect the probability of a word following the other, saved it to the database. This data base helped us to predict what word would follow another.

In the third phase, we used our predictor in order to improve the accuracy of our system. We first identified the uncorrected words, and found the word previous to it. Then we use the word previous to the wrong word and compared it with the earlier made database of our predictor. All words corresponding to the entered word would be displayed enabling us to determine what word should be used.

The system can so far recognize the basic slang words and convert it into the corresponding formal English words. Although further filtration and more accurate database can be created in order to deal with a vaster arena of slangs and people using it.

Future work:

In the future, normalization can be performed on multilingual languages, for eg. English and hindi or a mixture of both languages. Moreover, methods can be followed to improve efficiency or the system. The scope of this field is very large, hence efforts can be made to get better result with better precision.

6. REFERENCES

1. Nikola Ljubesi, Tomaz Eriavec, Daria Fiser; “Standardizing Tweets with Character-Level Machine Translation”; 15th International Conference, CICLing 2014, Kathmandu, Nepal, April 6-12, 2014, Proceedings, Part II; pp 164-175.
2. Swaran Lata and Somnath Chandra ; “Challenges of Multilingualism and Possible Approach for Standardization of e-Governance”, Department of Information Technology, Ministry of Communications & Information Technology, New Delhi, India ; pp 42-52.
3. Michael Carl ; Martin Kay ; Kristian Tangsgaard Hvelplund Jensen ; “Long Distance Revisions in Drafting and Post-editing”; CICLing-2010, Iasi, Romania, March 21-27, 2010; <http://openarchive.cbs.dk/bitstream/handle/10398/8046/LonDistRevision.pdf?sequence=1>
4. FranÇois Yvon; “Rewriting the orthography of sms messages”; Journal- Natural Language Engineering; Volume 16 Issue 2, April 2010; pp 133-159
5. Alistair Baron and Paul Rayson; “Automatic standardization of texts containing spelling variation”; Proceedings of the Corpus Linguistics Conference. Lancaster : Lancaster University; 2009; 25 p.