Vision Plan

For KDD- Service based Numerical Entity Searcher

(KSNES)

Version 1.1

Submitted in partial fulfillment of the Masters of Software Engineering Degree.

Naga Sowjanya Karumuri
CIS 895 – MSE Project
Department of Computing and Information Sciences
Kansas State University

Committee Members
Dr. William. H. Hsu
Dr. Torben Amtoft
Dr. Mitchell Neilsen
## Change Log

<table>
<thead>
<tr>
<th>Version #</th>
<th>Changed By</th>
<th>Release Date</th>
<th>Change Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version 1.0</td>
<td>Sowjanya</td>
<td>02/03/09</td>
<td>Initial Release</td>
</tr>
<tr>
<td>Version 1.1</td>
<td>Sowjanya</td>
<td>02/16/09</td>
<td>Added new definitions and references. Updating the diagrams</td>
</tr>
</tbody>
</table>
List of Figures

Figure 1: Project Overview.................................................................................................6
Figure 2: KSNES Data Flow Diagram.................................................................................9
Figure 3: System Use Case.................................................................................................10
Table of Contents

Change Log.........................................................................................................................2

List of Figures.........................................................................................................................3

1. Introduction.......................................................................................................................6
   1.1. Motivation.....................................................................................................................6
   1.2. NLP.............................................................................................................................6
   1.3. KDD- Service based Numerical Entity Searcher.........................................................7
   1.4. Terms and Definitions...............................................................................................7
   1.5. References..................................................................................................................8

2. Project Overview...............................................................................................................9
   2.1. Introduction.................................................................................................................9
      2.1.1. Stanford POS Tagger...........................................................................................9
      2.1.2. Numerical Phrase Extractor...............................................................................9
      2.1.3. Number-Unit/Date Pattern Recognizer.............................................................9
   2.2. Project Goal...............................................................................................................10
   2.3. Project Purpose.........................................................................................................10

3. Project Requirements.....................................................................................................11
   3.1. Application Requirements.......................................................................................12
      3.1.1. ARI 1 [Critical Requirement] ............................................................................12
      3.1.2. ARI 2..................................................................................................................12
      3.1.3. ARI 3 [Critical Requirement] ............................................................................12
      3.1.4. ARI 4..................................................................................................................12
      3.1.5. ARI 5 ..................................................................................................................12
      3.1.6. ARI 6 [Critical Requirement] ............................................................................12
      3.1.7. ARI 7 [Critical Requirement] ............................................................................13
      3.1.8. ARI 8..................................................................................................................13
      3.1.9. ARI 9 ..................................................................................................................13
3.1.10. ARI 10 [Critical Requirement] .................................................................13

3.2. POS Tagger Requirements ........................................................................13
  3.2.1. PTRI 1 [Critical Requirement] ..............................................................13
  3.2.2. PTRI 2 [Critical Requirement] ..............................................................13
  3.2.3. PTRI 3 .................................................................................................14
  3.2.4. PTRI 4 .................................................................................................14
  3.2.5. PTRI 5 [Critical Requirement] ..............................................................14
  3.2.6. PTRI 6 [Critical Requirement] ..............................................................14

3.3. Numerical Phrase Extractor Requirements .............................................14
  3.3.1. NPERI 1 [Critical Requirement] ............................................................14
  3.3.2. NPERI 2 ...............................................................................................14
  3.3.3. NPERI 3 ...............................................................................................14
  3.3.4. NPERI 4 [Critical Requirement] ............................................................15
  3.3.5. NPERI 5 [Critical Requirement] ............................................................15

3.4. Number-Unit/Date Pattern Recognizer Requirements .........................15
  3.4.1. NDPRRI 1 [Critical Requirement] ........................................................15
  3.4.2. NDPRRI 2 ............................................................................................15
  3.4.3. NDPRRI 3 ............................................................................................15
  3.4.4. NDPRRI 4 [Critical Requirement] ........................................................15
  3.4.5. NDPRRI 5 [Critical Requirement] ........................................................16

4. Assumptions .................................................................................................16

5. Constraints ....................................................................................................16

6. Environment ..................................................................................................16
1. Introduction

1.1. Motivation

The motivation of the project is to develop a system that can recognize the numerical and temporal patterns in text. Learning and understanding the numbers and time related information in natural text is important in tasks that attempt to recognize semantic entailment in pairs of sentences.

Apart from this, in most of the articles, as mentioned in [5], the occurrence of events is naturally anchored in time within the narrative text. For example, certain questions like – Is Bush currently the President of America? When was India attacked by Pakistan in last century? And many other questions like this are referred by the temporal aspects of the properties of the entities being questioned. Hence it is essential to be able to extract the timestamps of the events so as to understand the changes in the world over time and order the events with respect to one another.

Similarly, it is also important to know the quantity of entities along with the temporal information. For example, certain questions like – By how many votes did Obama win the elections? How many Oscar awards are won by Steven Spielberg? What was the highest temperature recorded in the year 2008? To answer such questions and maintain statistics over the entities, it is required to save information about the numbers that appear in the news.

Named Entity Recognition (NER) helps in identifying names of persons, locations, organizations and products in text but to understand the timestamps, ordering and persistence of events it is necessary to have a Natural Language Processing (NLP) system that can extract the numerical and temporal information from natural text.

1.2. NLP

Natural language processing (NLP) is a field of computer science concerned with the interactions between computers and human (natural) languages [8]. The term natural language is used to distinguish human languages (such as Spanish, Swahili or Swedish) from formal or computer languages (such as C++, Java or JSP). NLP may encompass both text and speech, but the current project involves text only. Some research areas in NLP are sentence understanding, probabilistic parsing and tagging, biomedical information extraction, grammar induction, word sense disambiguation and automatic question answering.

The present project belongs to the area of probabilistic parsing and tagging where a given sentence is POS tagged and then shallow parsed to extract the numerical information from the given sentence.
1.3. KDD- Service based Numerical Entity Searcher (KSNES)

The KSNES is such a system that should be able to identify the numerical and temporal information from natural text. The user should be able to interact with the system by entering the text in the textbox on the webpage. KSNES absorbs this text and outputs the number-unit/date on the webpage for the user to read.

![Figure1: Project Overview](image)

1.4. Terms & Definitions

**Actor** - In UML, an actor is the end user of the system.

**Entity** - Information that have real existence and are distinct and independent that are presented in the examples of section 1.1.

**Data Flow Diagram** – It is used for the visualization of data processing, how data flows in a system [8].

**IE** – Information Extraction [8]. It is a type of information retrieval whose goal is to automatically extract structured information, i.e. categorized and contextually and semantically well-defined data from a certain domain, from unstructured documents.

**IR** – Information Retrieval [8]. It is the science of searching for documents, for information within documents and for metadata about documents, as well as that of searching relational databases and the World Wide Web.

**Knowledge Discovery in Databases (KDD)** – A group headed by Dr. William H. Hsu whose primary focus is NLP.

**NABC** – National Agricultural Bio-Security Center

**NER** - Named Entity Recognition [8]. It is a subtask of information extraction that seeks to locate and classify atomic elements in text into predefined categories such as the names of persons, organizations, locations and miscellaneous entities.

**NLP** – Natural Language Processing [8]. It is a field of computer science concerned with the interactions between computers and human (natural) languages.

**Numerical patterns** - The phrases that represent numbers. Ex: 33 dollars, 1000 people, 145 miles, 1034.54, 843 tons etc., [1].
POS – Parts-of-Speech.
Penn Treebank Tag set – This acts as a repository for all the tags and their meanings that are used by the POS Tagger [6].
Shallow Parsing/Chunking – It is an analysis of a sentence which identifies the constituents (noun groups, verbs, verb groups, etc.) but does not specify their internal structure, or their role in the main sentence. In this project the chunking is done to identify the numerical phrases [8].
Sub-Chunker – A module of the project that takes the numerical/temporal phrases as input and outputs the number-unit/date.
Temporal patterns - The phrases that denote date/time. Ex: Dec 10th 1986, 2009, 5 May, this year, today, Monday, January etc., [1].
Textual Entailment – It is the relation between two sentences where the truth of one requires the truth of the other.
Unified Modeling Language (UML) – A standard notation used to describe real-world objects.
Use Case Diagram – A behavioral diagram defined by UML. It provides a graphical depiction of system functionality in terms of actors.

1.5. References


2. Project Overview

This section presents the information about the overall view of the working of the system and introduces the modules that made the KSNES work.

2.1. Introduction

The block diagram in Figure 2 gives a bird’s view of the internal working of the system along with an example that shows the data at the end of each module in the system. User should be entering the text that is required to be chunked through a webpage.

2.1.1. Stanford POS Tagger

The text entered by the user is given to the Stanford Log-linear POS tagger that can tokenize and tag the sentence with POS such as noun, verb, adjective etc., which is developed by Stanford NLP Group. The words in the sentence are tagged by using the Penn Treebank tag set which has notations for the tags that should be assigned to words in different contexts. For example, if the input text is: I lost 33 dollars in 1999. Then the sentence would be tagged by the POS Tagger as – I/PRP lost/VBD 33/CD dollars/NNS in/IN 1999/CD. Here PRP (Personal pronoun), VBD (Verb, past tense), CD (Cardinal number), NNS (Plural noun) are some of the tags from the Penn Treebank tag set which has lot more such tags like those that describe conjunctions (and, but, or), prepositions (of, in), cardinal numbers (1, 16, 26, three), personal pronouns (I, you), possessive pronoun (your, one’s) etc.,

2.1.2 Numerical Phrase Extractor

The tagged output from the POS Tagger will be fed to a Numerical Phrase Extractor which filters numerical phrases from the tagged sentence. This part of the system should be able to identify the dates, the numbers, units associated with them and type to which the unit belongs. From the example discussed in the previous section, ‘33 dollars’ and ‘1999’ should be extracted using the shallow parsing technique.

2.1.3 Number-Unit/Date Pattern Recognizer

The extracted numerical phrase will be fed to the Number-Unit/Date Pattern Recognizer that should be able to present the number-unit/date information of the given textual input. The output for the above mentioned example would then look like this - ‘33 dollars’: type: number, value: 33, unit – dollars, unit-type – money; ‘1999’: type: date, value: 0/0/1999. Here ‘type’ specifies if it’s a number or a temporal phrase, value is the number or the date, unit is the physical unit like miles, dollars, cubic-meters etc., unit-type is the category of the unit like distance, currency, volume etc.,
Thus the required output is generated and displayed to the user on the webpage.

Figure 2: KSNES Data Flow Diagram
2.2. Project Goal

The primary focus of KSNES is to construct a system that should be able to extract the numerical data from natural language text and hence can also be called as Numerical Quantifier. The project is service based where the whole system is put on the web server and is access using a GUI which is a webpage. There are no intermediate steps for the user to perform after sending the input from the main webpage by clicking the send button; user should be able to collect the output on the same page.

2.3. Project Purpose

The present application is developed for the purpose of providing a good base for the KDD students to perform numerical sub-chunking. The KDD group members can use this project to collect some statistical data like finding the outbreak of the disease, number of cattle affected etc., from the web crawled data in the NABC related project. This project since made a service can be interfaced by other programs to perform numerical sub-chunking.

3. Project Requirements

In this section, we will discuss the critical requirements and other requirements that should be satisfied to accomplish a working KSNES system. Each requirement is given a unique requirement number and discussed in detail. The critical requirements in the project will be noted near the description of the requirement and the planned release that will fulfill that requirement is also mentioned.

Figure 3 is the use case diagram that visualizes the actions that a user performs to interact with the system. Here we can see that the user opens the webpage, enters the text and clicks on the send button. The request invokes the POS tagger that is set on the server which in turn corresponds with the numerical phrase extractor and Number-Unit/Date Pattern Recognizer which send the output to be displayed back on the webpage. These are the major functions of the user.

![Figure 3: System Use Case](image-url)
The requirements explained in the below sections are those that are to be fulfilled by the final system. They are divided based on the different modules that the system is composed of. They are – Application Requirements, POS Tagger requirements, Numerical Phrase Extractor, Number-Unit/Date Pattern Recognizer. Modularizing the project helps to do updates and modifications to different modules of the project without affecting other modules. It will also make the modules of the system reusable and easy to understand.

3.1. Application Requirements

   The overall idea of the system is explained in the requirements specified in this section. The modules that are unseen to the user are explained in the coming sections. The requirements are numbered as ARI X, where ARI stands for Application Requirement Item.

3.1.1. ARI 1 [Critical Requirement]
   The program shall provide a GUI for the user interaction. This is a critical requirement because the usefulness of the system would be extremely limited and complicated if done in a command line format.
   - Build Release Applicability: Demo2, Final Release

3.1.2. ARI 2
   The application shall be executable in a single step and no user interaction is required.
   - Build Release Applicability: Demo2, Final Release

3.1.3. ARI 3 [Critical Requirement]
   The application shall be started when the user enters the text into the text box in the webpage. This is a critical requirement because the input given by the user is processed and modified by application to give the final output.
   - Build Release Applicability: Demo2, Final Release

3.1.4. ARI 4
   The application shall invoke the other modules of the project when the user clicks on the send button.
   - Build Release Applicability: Demo2, Final Release

3.1.5. ARI 5
   The application shall clear the text in the text box when the clear button is clicked by the user.
   - Build Release Applicability: Demo2, Final Release

3.1.6. ARI 6 [Critical Requirement]
   The GUI and the POS tagger shall be platform independent and the Numerical Extractor and the Number-Unit/Date Pattern Recognizer shall work successfully only on the Linux and Unix machines. This is a critical requirement because the
application is being developed using windows and Linux to make different modules work collectively.

- **Build Release Applicability:** Final Release

3.1.7. **ARI 7 [Critical Requirement]**
The user shall be able to view the chunked output on the webpage. This is a critical requirement because the application’s goal is to show the output to the user.

- **Build Release Applicability:** Final Release

3.1.8. **ARI 8**
The user shall be able to stop the running of the application after viewing his output by closing the web browser.

- **Build Release Applicability:** Final Release

3.1.9. **ARI 9**
The user shall be able to run the application again with a new input ones the previously entered text is chunked.

- **Build Release Applicability:** Final Release

3.1.10. **ARI 10 [Critical Requirement]**
The user shall be able to enter the input of any size. This is critical because the application shall not restrict the user’s freedom on the size of the input the user can to enter.

- **Build Release Applicability:** Demo2, Final Release

3.2. POS Tagger Requirements

This section deals with the working of the tagger. This module of the project does not interact with the user but is still being mentioned so as to understand the underlying working of this module. The requirements are numbered as PTRI X, where PTRI stands for POS Tagger Requirement Item.

3.2.1. **PTRI 1 [Critical Requirement]**
The POS Tagger shall be given the raw input text that user enters on the webpage. This is a critical requirement because the input is required to be tagged with POS to be processed further.

- **Build Release Applicability:** Demo2, Final Release

3.2.2. **PTRI 2 [Critical Requirement]**
This POS Tagger shall be a service request from the main webpage. This is critical because POS tagger is a third party segment and can be run efficiently when used as a service.

- **Build Release Applicability:** Demo2, Final Release
3.2.3. **PTRI 3**  
The POS Tagger shall be able to tokenize the given text.  
- **Build Release Applicability:** Demo2, Final Release

3.2.4. **PTRI 4**  
The POS Tagger shall be able to tag the words, punctuations and symbols in the sentence using the Penn Treebank Tag set.  
- **Build Release Applicability:** Demo1, Demo2, Final Release

3.2.5. **PTRI 5 [Critical Requirement]**  
The POS Tagger shall be able to produce the tagged sentence. This is a critical requirement because the module should be producing the expected output.  
- **Build Release Applicability:** Demo1, Demo2, Final Release

3.2.6. **PTRI 6 [Critical Requirement]**  
The POS Tagger shall be able to send the tagged sentence to the next module which is the Numerical Phrase Extractor. This is a critical requirement because the module should be passing the information for the KSNES to work further.  
- **Build Release Applicability:** Final Release

3.3. Numerical Phrase Extractor Requirements

This section presents the requirements of the phrase extractor which takes in the tagged text and filters out the numerical phrase. This module of the project does not interact with the user but is still being mentioned so as to understand the underlying working of the module. The requirements are numbered as NPERI X, where NPERI stands for Numerical Phrase Extractor Requirement Item.

3.3.1. **NPERI 1 [Critical Requirement]**  
The Numerical Phrase Extractor shall be able to take the tagged sentence from the POS tagger. This is a critical requirement because this module requires the output of the previous module to proceed further in chunking.  
- **Build Release Applicability:** Final Release

3.3.2. **NPERI 2**  
The Numerical Phrase Extractor shall be able to identify the tagged words that may be containing the numbers and the units.  
- **Build Release Applicability:** Demo2, Final Release

3.3.3. **NPERI 3**  
The Numerical Phrase Extractor should be able to identify the tagged words that may be containing the dates.  
- **Build Release Applicability:** Demo2, Final Release
3.3.4. **NPERI 4 [Critical Requirement]**
The Numerical Phrase Extractor should be able to produce the filtered number-unit or the date phrase. This is a critical requirement because the module should be producing the expected output.
- **Build Release Applicability:** Demo2, Final Release

3.3.5. **NPERI 5 [Critical Requirement]**
The Numerical Phrase Extractor should be able to send the filtered phrase to the next module which is the Number-Unit/Date Pattern Recognizer. This is a critical requirement because the module should be passing the information for the KSNES to work further.
- **Build Release Applicability:** Final Release

3.4. Number-Unit/Date Pattern Recognizer Requirements

This module takes the extracted numerical phrase and determines the number, unit, date, unit-type in it and presents it to the user on the webpage. This module of the project does not interact with the user but is still being mentioned so as to understand the underlying working of the module. The requirements are numbered as NDPRRI X, where NDPRRI stands for Number-Unit/Date Pattern Recognizer Requirement Item.

3.4.1. **NDPRRI 1 [Critical Requirement]**
The Number-Unit/Date Pattern recognizer shall be able to take the extracted phrase from the Numerical Phrase Extractor. This is a critical requirement because this module requires the output of the previous module to proceed further in chunking.
- **Build Release Applicability:** Final Release

3.4.2. **NDPRRI 2**
The Number-Unit/Date Pattern recognizer shall be able to identify the numbers, units and unit-type if present in the phrase.
- **Build Release Applicability:** Demo1, Demo2, Final Release

3.4.3. **NDPRRI 3**
The Number-Unit/Date Pattern recognizer shall be able to identify the date, month and the year if present in the given phrase.
- **Build Release Applicability:** Demo1, Demo2, Final Release

3.4.4. **NDPRRI 4 [Critical Requirement]**
The Number-Unit/Date Pattern Recognizer shall be able to produce the number, unit corresponding to it and the type to which the unit belongs to if it’s a number phrase. And if it is a temporal phrase then the module should be able to display the date based on the number, month and year information in the phrase. This is a critical requirement because the module should be producing the expected output.
- **Build Release Applicability:** Demo1, Demo2, Final Release
3.4.5. NDPRRI 5 [Critical Requirement]
The Number-Unit/Date Pattern Recognizer shall be able to display the value, unit, and unit-type to the user on the webpage. This is a critical requirement because the module should be able to produce the output since it’s the final output expected output of the KSNES system.
- **Build Release Applicability**: Final Release

4. Assumptions

- The user will have to interact with the system using webpage.
- The system requires Java 1.5 or later to be installed on the computer running the application.
- The user will need a minimum of 512 MB of memory.
- The user will need to have a computer with a minimum speed of 1.6 GHz.

5. Constraints

- POS Tagger is not implemented as a part of this project. A Stanford Log-linear POS Tagger, developed by Stanford University NLP Group, is used that takes the raw text and provides a POS tagged sentence.
- Numerical Quantizer developed by Ben Sapp, UIUC is used as a base to develop the Number-Unit/Date Pattern Recognizer that outputs the number-unit/date information of the given numerical phrase.
- The whole system should be set up as a service on a server so that anyone can interface to it or use it.

6. Environment

- Eclipse 3.3.0 will be used to develop the Numerical Phrase Extractor.
- Java version 1.5 will be used to run the POS Tagger and it should be setup as a service.
- Version control will be handled using Tortoisesvn.
- Number-Unit/Date Pattern Recognizer will be developed using GNU C++.