PONTIFICAL CATHOLIC UNIVERSITY OF RIO GRANDE DO SUL
FACULTY OF INFORMATICS
COMPUTER SCIENCE GRADUATE PROGRAM

A PROPOSAL FOR AN ARCHITECTURE TO EXTRACT INFORMATION FROM SMS MESSAGES DURING EMERGENCY SITUATIONS

DOUGLAS MACHADO MONTEIRO

Thesis submitted to the Pontifical Catholic University of Rio Grande do Sul in partial fulfillment of the requirements for the degree of Master in Computer Science.

Advisor: Prof. Vera Lúcia Strube de Lima

Porto Alegre
2015

Dados Internacionais de Catalogação na Publicação (CIP)

M775p Monteiro, Douglas Machado

A proposal for an architecture to extract information from SMS messages during emergency situations / Douglas Machado Monteiro. – Porto Alegre, 2015.
123 p.

Dissertação (Mestrado) – Faculdade de Informática, PUCRS.
Orientador: Profª. Drª. Vera Lúcia Strube de Lima.

1. Informática. 2. Processamento da Linguagem Natural. 3. Recuperação da Informação. I. Lima, Vera Lúcia Strube de. II. Título.

CDD 006.35

Ficha Catalográfica elaborada pelo
Setor de Tratamento da Informação da BC-PUCRS

Pontifícia Universidade Católica do Rio Grande do Sul
FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

TERMO DE APRESENTAÇÃO DE DISSERTAÇÃO DE MESTRADO

Dissertação intitulada "*A Proposal for an Architecture to Extract Information from SMS Messages During Emergency Situations*" apresentada por Douglas Machado Monteiro como parte dos requisitos para obtenção do grau de Mestre em Ciência da Computação, aprovada em 16/03/2015 pela Comissão Examinadora:

Vera Lúcia Strube de Lima

Profa. Dra. Vera Lúcia Strube de Lima– Orientadora — PPGCC/PUCRS

Prof. Dr. Duncan Dubugras Alcoba Ruiz – — PPGCC/PUCRS

Prof. Dr. Leandro Krug Wives– — UFRGS

Homologada em 23/04/2015, conforme Ata No. 006 pela Comissão Coordenadora.

Prof. Dr. Luiz Gustavo Leão Fernandes
Coordenador.

PUCRS
Campus Central
Av. Ipiranga, 6681 – P32– sala 507 – CEP: 90619-900
Fone: (51) 3320-3611 – Fax (51) 3320-3621
E-mail: ppgcc@pucrs.br
www.pucrs.br/facin/pos

Dedico este trabalho a minha noiva Mônica.

“Intelligence without ambition is a bird without wings.”
(Salvador Dali)

ACKNOWLEDGMENTS

Agradeço com amor a Mônica, pelo carinho, incentivo, companheirismo, paciência e, compreensão nos momentos em que precisei me distanciar para concluir esta pesquisa. Obrigado por me inspirar a superar meus limites.

Agradeço aos meus pais, por sempre me incentivarem a estudar e correr atrás de meus sonhos, apoiando todas as minhas decisões.

Agradeço a todos os professores da PUCRS, que compartilharam conhecimento comigo, fazendo com que eu enxergasse além do cotidiano de minhas atividades profissionais e preparando-me para novos desafios.

Agradeço em especial a minha orientadora, Vera Lúcia Strube de Lima, que sempre foi bastante solícita, me apoiando, incentivando e confiando em minhas capacidades, além de me motivar a buscar sempre o melhor de mim.

Agradeço ao meu amigo Rodrigo Martins, pelos conselhos, contribuições e por ser um dos principais incentivadores para que eu ingressasse na carreira acadêmica.

Agradeço aos colegas de trabalho, pela contribuição com apoio e sugestões para a realização deste projeto.

Agradeço a CAPES por ter financiado meus estudos.

PROPOSTA PARA UMA ARQUITETURA DE EXTRAÇÃO DE INFORMAÇÃO DE MENSAGENS SMS DURANTE SITUAÇÕES DE EMERGÊNCIA

RESUMO

Durante situações de emergência, uma grande quantidade de informação é trocada via mensagens SMS. Estas mensagens costumam ter escrita informal e contêm abreviações e erros de grafia, o que dificulta seu processamento. Este é um problema para as ferramentas de Extração de Informação atuais, especialmente para o Português. Este trabalho propõe uma arquitetura de extração de informação de mensagens SMS em situações de emergência. A arquitetura contempla quatro componentes: processamento linguístico, processamento temporal, processamento de eventos e fusão da informação. Também se define um processo para criação de corpus de SMSs. A partir da arquitetura proposta, foi realizado um estudo de caso que incluiu a construção do BraCorpSMS, um corpus de mensagens SMS recebidos por uma companhia de energia elétrica e um protótipo em Python utilizando NLTK para validar a arquitetura. O protótipo teve seus componentes de Extração de Informação avaliados, obtendo 88% de Precisão, 59% de Cobertura e 71% de Medida-F. Os resultados indicam oportunidades de avanços, mas, sendo este o primeiro trabalho para o Português voltado para o processamento de mensagens SMS em situações de emergência, também serve de roteiro para trabalhos futuros nesta área.

Palavras-Chave: extração de informação, mensagens curtas, construção de corpus de mensagens curtas, emergências.

A PROPOSAL FOR AN ARCHITECTURE TO EXTRACT INFORMATION FROM SMS MESSAGES DURING EMERGENCY SITUATIONS

ABSTRACT

In mass emergencies, a fair amount of information is exchanged via SMS messages. These messages tend to be informal and to contain abbreviations and misspellings, which makes them difficult to treat. This is a problem for current Information Extraction tools, especially for messages in Portuguese. This work proposes an architecture to extract information from SMS messages during emergencies. The architecture comprises four components: Linguistic Processing, Temporal Processing, Event Processing, and Information Fusion. We also defined an SMS corpus building process. From the proposal of this architecture, we conducted a case study, which included building BraCorpSMS, a corpus of SMS messages received by an electric utility company. We built a prototype in Python using NLTK to validate the architecture. The prototype had its Information Extraction components evaluated achieving Precision of 88%, Recall of 59% and balanced F-measure of 71%. The results indicate improvement opportunities, but as this is the first work for Portuguese facing processing SMS messages during emergency situations, it also serves as a roadmap for future work in the area.

Keywords: information extraction, short messages, short message corpus building, emergencies.

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

ACE – Automatic Content Extraction

CSV – Comma Separated Value

DRPU – Durative, Relative, Precise and Unique

HMM – Hidden Markov Model

IE – Information Extraction

IR – Information Retrieval

ISO – International Organization for Standardization

LREC – Language Resources and Evaluation Conference

NE – Named Entity

NER – Named Entity Recognition

NILC – Núcleo Interinstitucional de Linguística Computacional

NLP – Natural Language Processing

NLTK – Natural Language Toolkit

POS – Part-of-Speech

PRFU – Punctual, Relative, Fuzzy and Unique

PRPU – Punctual, Relative, Precise and Unique

RDF – Resource Description Framework

SPARQL – SPARQL Protocol and RDF Query Language

SMS – Short Message Service

SVM – Support Vector Machine

TE – Temporal Expression

TER – Temporal Expression Recognition

TIMEX – Time Expression

XML – Extensible Markup Language

CONTENTS

1. INTRODUCTION

Natural Language Processing (NLP) plays a strategic role in Information Extraction applications. Combining NLP resources with large knowledge bases can improve documents search and retrieval systems. Areas such as Bioinformatics, for example, have employed these resources, using sets of ontologies to combine text elements with concepts and with other texts [15].

Recently, Document Retrieval and Information Extraction (IE) have become some of the most popular applications in the natural language processing area. This process is made possible by understanding the information contained in texts and their context, and is a very complex task, particularly when treating texts from SMS messages, tweets and messages in other social networks. Users of this service write messages freely, with abbreviations, slangs, and misspellings. Current tools face difficulties when processing such informal language [30, 3, 46].

These challenges have attracted the interest of researchers, as seen in [8, 46]. Short messages tend to be brief, informal and to present similarities to speech. Messages using the Short Message Service (SMS) are widely used for numerous purposes, which makes them rich and useful data for information extraction. The content of these messages can be of high value and strategic interest, especially during emergencies[1]. Under these circumstances, the amount of messages tends to increase considerably. Moreover, SMS messages contain valuable information employable in service provision and may promote greater agility and precision when meeting demands.

1.1 Research Problem

Currently, it is hard to imagine any line of business that does not use any textual information. According to Grant Ingersoll *et al.* [17], the average time spent by a worker using e-mail is 13 hours per week. Besides e-mail, there are still social networks, instant messaging and many other communication channels that make men use about 9 hours a week searching for information and another 8 hours analyzing it [17].

Aside from the already available content on the Internet, on a daily basis, a large amount of new information are generated and transmitted by computers around the world. Gary Miner *et al.* [31] estimate that 80% of the information available in the world are in free text format and therefore not structured. To understand such content, it is necessary to analyze it completely. With such large amount of potentially relevant data, an Information

[1] Also referred to as crisis events, disasters, mass emergencies and natural hazards by other researchers in the area.

Extraction system can structure and refine raw data "to find and link relevant information while ignoring extraneous and irrelevant information" [9].

However, in the midst of this scenario, there are many obstacles, mainly considering instant messaging. There are many difficulties in rapidly gathering short messages as well as in building a corpus from SMS messages, since private companies tend not to make them publicly available [34, 55]. Besides, the content of SMS messages is often personal, raising confidentiality issues. Hence, only a few available SMS corpora are found [6]. Thus, in order to study the communication via SMS messages during emergencies, we needed to build a corpus composed of messages under these circumstances.

1.2 Contributions

After an extensive analysis of the area, we propose an architecture to extract information contained on incoming SMS messages during emergency situations. This IE architecture has as input a corpus of SMS messages and comprises four components: a Linguistic Processing component, a Temporal Processing component, an Event Processing Component, and an Information Fusion component. The Linguistic Processing component receives an SMS Corpus and is responsible for preprocessing messages. It handles with abbreviations and punctuation, sentence splitting, tokenization and stopword removal. The Temporal Processing Component uses rules and a list of temporal keywords to identify and classify temporal expressions. The Event Processing Component is responsible for identifying events according to a set of domain-defined categories and it provides additional information regarding situation awareness, such as fallen trees, power outages, lightning strikes, etc. As output, the architecture consolidates information graphically, grouping crisis events according to their severity degrees, notification types, and temporal information.

From this architecture, one should obtain strategic information in a human understandable form, in order to help the decision-making process. In addition, to the best of our knowledge, there is no architecture to address this matter, especially when considering the Portuguese language. We expect this proposal to bring focus to this area and encourage other researchers to contribute to its improvement.

The Information Extraction Architecture depends on an input corpus, which should be rapidly collected, anonymized, normalized and prepared for the application of part-of-speech taggers, time analyzers, and event taggers. Besides, each new episode requires a new execution of the corpus building procedure, having in mind crisis events analysis and information extraction. Consequently, in this work, as there is no defined process for this purpose, we propose an SMS corpus building process. The input of the corpus building process is a set of raw short messages. The process comprises five steps, namely data

collection, data filtering, anonymization, normalization, and formatting and its output is an SMS corpus.

1.3 Text Organization

This dissertation is organized in six chapters, the first one being this introduction, which presented the context and motivation of the work, the research problems, and its contributions. In Chapter 2, we expand this context presenting a review on fundamentals in the area of Information Extraction and comment on necessary linguistic resources towards Information Extraction from texts. Chapter 3 concludes this review by presenting related work on SMS Corpora Building, Information Extraction from short messages and its applications.

In Chapter 4, we introduce the proposal for an SMS Information Extraction architecture that is turned to message exchanged during emergencies. Together with this architecture, we also propose a SMS Corpus Building process, as IE applications based on this architecture will need and input corpus built from messages exchanged in this scenario and new corpora must be built for each new IE application run. Chapter 5 provides details of a case study conducted in order to validate this architecture over BraCorpSMS, a corpus built from SMS Messages sent by costumers to an electric utility company during emergencies. Finally, in Chapter 6, we make considerations about results and comment on challenges faced, as well as on future work.

2. FUNDAMENTALS IN THE AREA

This chapter is organized in 5 sections, where in Section 2.1 we present the main information extraction concepts and techniques related to this research. From Section 2.2 to Section 2.3, we introduce the concept and features of corpus and concerns regarding the use of corpora in Information Extraction. In Section 2.4, we highlight some of the most used evaluation metrics for NLP systems. Finally, in Section 2.5 we review relevant points expressed during this chapter and comment on their relation with the present work.

2.1 Information Extraction

Here, we present concepts and techniques related to Information Extraction, based mainly on Speech and Language Processing, a book written by Jurafsky and Martin [20], on the Oxford Handbook of Computational Linguistics, by Ruslan Mitkov [32] and on The Handbook of Computational Linguistics and Natural Language Processing, by Alexander Clark *et al.* [7].

According to Cowie and Lehnert [9], "with large amounts of potentially useful information in hand, an IE system can then transform the raw material, refining and reducing it to a germ of the original text". The Information Extraction process "transforms unstructured information, within texts, in structured data" [20]. Once structured, one can analyze such data. According to Jurafsky and Martin [20], robust information extraction approaches comprise a combination of existing techniques. Finite state automata, probabilistic models, syntactic chunking and machine learning (ML) form the core of most IE approaches on tasks such as Named Entity Recognition (NER), Temporal Expression Recognition and Event Detection and Classification.

2.1.1 Named Entity Recognition

Named Entity Recognition (NER) usually covers detecting and classifying proper names in a text. It is the first step for most IE tasks. In general, systems that recognize named entities tend to focus on finding persons names, places and organizations mentioned in news [32].

Depending on the context, entities may have a certain meaning that sets them apart from ordinary text. For instance, while processing the standard Portuguese language, one may suggest that two adjacent capitalized words in a text compose a proper name. If ‘Dr.’ precedes these words, then it is likely that this is a person’s name. However, if these words

are preceded by ‘arrived in’, they refer to a location. Classifying a word as a proper name is specifically context-dependent. While generic NER systems detect people, places and organizations, particular applications may have interest in detecting other kind of entities, such as molecules, medicines or commodities. NER applications often use resources such as lists of proper nouns, and techniques as rule-based matching and supervised machine learning [20, 7].

Hand-coded Rules

Rules are connections between a condition and the resultant action. The condition-action paradigm comprises recognizing a token or sequence of tokens (the condition), and then applying a tag (the action). Domain-constrained recognition tasks such as finding event expressions, identifying temporal expressions and NEs often use rule-based systems [40]. According to Clark *et al.* [7], the first NER systems were based on hand-coded rules, which are regular expressions. The part of the text that matches the rule is tagged, indicating a NE, as shown in Table 2.1.

Table 2.1 – Regular Expressions used in NER [7]

Rule	**Tag**
(capitalized-token) + "Inc."	Organization
"Mr." [capitalized-token? initial? capitalized-token]	Person

In the examples in Table 2.1, ‘+’ indicates the existence of one or more instances of a token and ‘?’ indicates that the token is optional. In the former rule, tokens represent an organization name; in the latter, tokens inside the brackets represent a person’s name. Since it is common to mention an entity several times in a text, one can improve the precision of named entity recognition using a list of identified and classified names. One can use such practice in different situations, once the name has been identified in a context, facilitating the classification of other instances of the same name. According to Pustejovsky and Stubbs [40], rule-based systems are a good way to identify features that may be useful in a document without having to take the time to train an algorithm. For some tasks, like temporal expression recognition, rule-based systems outperform machine learning algorithms, according to the same authors.

Part-of-Speech Tagging

Part-of-speech (POS) tags can be used to label words according to their types. Assigning part-of-speech tags automatically is appropriate in any NLP task, such as word-sense disambiguation and shallow parsing of texts, to find names, dates, times or named

entities in information extraction [20]. Many algorithms have been applied to grammatical tagging, including rule-based tagging, probabilistic tagging with HMM and Maximum Entropy, among others [20].

Traditionally, POS tags are based on morphological and syntactic functions. Words that have a similar function regarding their distributional syntactic properties or their morphological properties are grouped in classes. Table 2.2 below shows the categories used in Floresta Sintá(c)tica[1], a treebank of POS tagged sentences in Portuguese, with their corresponding tags.

Table 2.2 – Examples of Floresta Tags

Category	**Tag**
Noun	n
Proper Noun	prop
Adjective	adj
Verb (infinitive, finite, participle, gerund)	v-fin, v-inf, v-pcp, v-ger
Article	art
Pronoun (personal, determined, independent)	pron-pers, pron-det, pron-indp
Adverb	adv
Numeral	num
Preposition	prp
Interjection	intj
Conjunction (subordinative, coordenative)	conj-s, conj-c

Commas, periods and other signs also receive specific tags. Using the tags exposed in Table 2.2, one can label the following sentence tokens as in Table 2.3, for sentences "Eu estou sem luz desde ontem" (There is no electricity since yesterday) and "Um vento forte derrubou a árvore" (A strong wind toppled the tree).

Table 2.3 – Examples of POS tagged sentences

Sentence	Eu	estou	desde	ontem	sem	luz	.
Tags	n	v-fin	prp	adv	prp	n	.
Sentence	Um	vento	forte	derrubou	a	árvore	.
Tags	art	n	adj	v-fin	art	n	.

2.1.2 Temporal Expression Recognition

Finding out when an event occurs is one of the main interests of event extraction. Temporal expressions recognition (TER) consists of standardizing groups of words or other characters that correspond to temporal expressions (TEs) in a text so a computer can

[1] http://www.linguateca.pt/floresta/principal.html

process them. Whilst absolute temporal expressions refer explicitly to specific times, such as dates or hours, relative temporal expressions are implicit references to time, as "daqui uma semana" (in a week). Additionally, durations indicate periods of time at different levels (seconds, hours, months, etc.). Tenses may also indicate past, present or future actions.

The construction of temporal expressions starts from lexical triggers, that can be nouns (morning, summer, dawn), proper names (January, Easter), adjectives (past, recent) or adverbs (daily, monthly) [20].

Temporal analysis aims to relate temporal expressions found to specific dates, times, and events that have occurred during this period comprising the following subtasks:

- Identifying temporal expressions according to a time or date;
- Associating temporal expressions to events found in the text;
- Sorting events according to when they occurred.

The syntactic construction of temporal expressions starts from a lexical trigger. Triggers can be nouns, proper nouns, adjectives or adverbs [20]. Table 2.4 presents some examples of lexical triggers.

Table 2.4 – Examples of lexical triggers (adapted from [20])

Category	**Examples**
Noun	morning, night, summer, dawn
Proper Noun	January, Easter, monday
Adjective	past, recent, ancient
Adverb	daily, monthly, yearly

The TimeML[2] annotation scheme standard provides a XML tag system - currently TIMEX3[3] - using triggers to tag temporal expressions. The example in Table 2.5 shows the use of this scheme regarding the temporal expression ’30 minutes’:

Table 2.5 – Example of a Timex3 Tag

Temporal Expression	**Value**	**TIMEX3 Tag**
30 minutes	PT30M	<TIMEX3 tid="t1" type="DURATION" value="PT30M">30 minutes</TIMEX3>

As there are different ways one can represent Temporal Expressions, it is essential to normalize them. Temporal normalization occurs after Temporal Expression Recognition and comprises mapping Temporal Expressions onto dates, times or periods. Frequently, Temporal Expressions found in texts are incomplete [20]. Most expressions such as in news reports are implicitly linked to the publication date, which is called a ‘temporal anchor’. From

[2]http://www.timeml.org/

[3]http://www.timeml.org/site/publications/timeMLdocs/timeml_1.2.1.html

an anchor, one can identify to which period expressions like "yesterday", "tomorrow" or "in two days" refer. Moreover, Temporal Expressions may anchor to events mentioned in a text, what makes it necessary to identify the date or time of occurrence of a certain event, before addressing the temporal expressions. Kevers [22] ranked Temporal Expressions according to 4 groups, which combined lead to 16 categories:

- Punctual (referring to a specific moment) or durative;
- Absolute or relative;
- Precise or fuzzy;
- Unique or repetitive.

According to this specification, one classifies “July 2nd, 2014” as a punctual, absolute, precise and unique expression, whereas “July 2nd, around 21h” is punctual, absolute, fuzzy and unique. Following Kever’s classification, Weiser *et al.* [58] indicate that, for French, the TE categories appearing on SMS messages are: PRPU (punctual, relative, precise and unique), DRPU (durative, relative, precise and unique) and PRFU (punctual, relative, fuzzy and unique). Besides these, Weiser *et al.* create yet another category for dialogues, which handles situations that will happen in the future, as in “see you tomorrow”.

According to Jurafsky and Martin [20], temporal expressions refer to absolute times, relative times, durations and sets of them. One can directly map absolute temporal expressions to calendar dates, times or both. Relative temporal expressions map specific times through a point of reference, as in “one week after last Tuesday”. Durations denote periods with different levels of granularity (seconds, minutes, days, weeks, etc.). Since TEs are usually limited to a set of patterns, TER systems in general are pattern-based.

2.1.3 Event Detection and Classification

The NE concept is commonly extended to "things" that are not necessarily entities but have practical importance and definable features that mark their presence. In most texts, one describes the events in which the named entities are involved. Discovering, analyzing and finding the relation between events in a text is essential to understand the subject of the text. Events usually refer to something that draws attention. Concerts, soccer matches and elections are examples of events.

According to Jackson and Moulinier [18], "unlike more ambitious forms of NLP, information extraction programs analyze only a small subset of any given text, *e.g.*, those parts that contain certain ’trigger’ words, and then attempt to fill out a fairly simple form that represents the objects or events of interest". Jurafsky and Martin indicate that "the task of

event detection and classification is to identify mentions of events in texts and then assign those events to a variety of classes. For the purposes of this task, an event mention is any expression denoting an event or state that can be assigned to a particular point, or interval, in time" [20]. Often, events are mentioned through verbs, but they can also be nouns, like ‘fire’ or ‘storm’, or even adjectives, when they indicate a state that has changed. However, not all nouns, verbs and adjectives are events, given that this definition depends on the context in which the word is used.

According to Clark *et al.* [7], from a certain scenario specification or event type, one must be able to identify occurrences of such event with their arguments and modifiers. Similar to NER, event detection addresses the problem of resolving references, *i.e.*, in a given text, identifying existing references of the same event.

One can use machine learning and rule-based approaches for event detection. Both approaches make use of surface information, like part-of-speech and verb tense information [20]. According to Wang *et al.* [56], several approaches were proposed for event detection on short messages, such as text classification and clustering.

2.2 Corpus Processing

This section presents the idea of corpus, types of corpora and some of the concerns regarding the use of corpus in NLP. The content displayed here is based on [31, 17, 20, 27, 40]. In addition, we mention some of the existing Portuguese corpora and the recognizable features of short message corpora.

2.2.1 Corpus

In the context of NLP, data volumes in text format are treated as linguistic corpora. Gary Miner *et al.* [31] define corpus as a large and structured set of texts (usually held and processed electronically) prepared for knowledge discovery purposes. It is worth mentioning the existence of speech corpora, consisting of speech recordings, usually containing phonetic representations, pronunciation variations and word fragments [27]. Considering text and speech corpora, Sardinha [45] states that "a corpus is a collection of spoken and written texts carefully collected in order to represent a language or language variety". In the present work, we focus on text corpora.

Manning and Schütze [27] suggest that the essential requirements to work with Natural Language Processing are computers, software and corpora. This means that, since the use of corpora is central to NLP and text corpora are usually large volumes of texts, the

computing power needed to handle a vast amount of information is a determining factor for processing corpora.

According to Jurafsky and Martin [20], a corpus is a collection of machine-readable texts that have been produced in a natural communicative environment. The sample that generates the corpus must be representative and balanced, according to the relevance of the information one wishes to observe. For example, preparing a sample by genre, containing various texts such as newspaper articles, fiction books, blogs, law articles, among others. Additionally, a corpus is said "representative of a variety of language" if its content allows generalization to that variety. In other words, this means that if the content extracted to build the corpus through specifications and studies reflects the larger part of its population, then it can "represent" this language.

Corpus Partitioning

According to the most widely used model for machine learning, the corpus is separated in two parts: the development corpus and the test corpus [40]. Subsequently, the developing corpus is divided into a training set and a development-test set. The data is distributed randomly between sets. Figure 2.1 illustrates this distribution of the corpus.

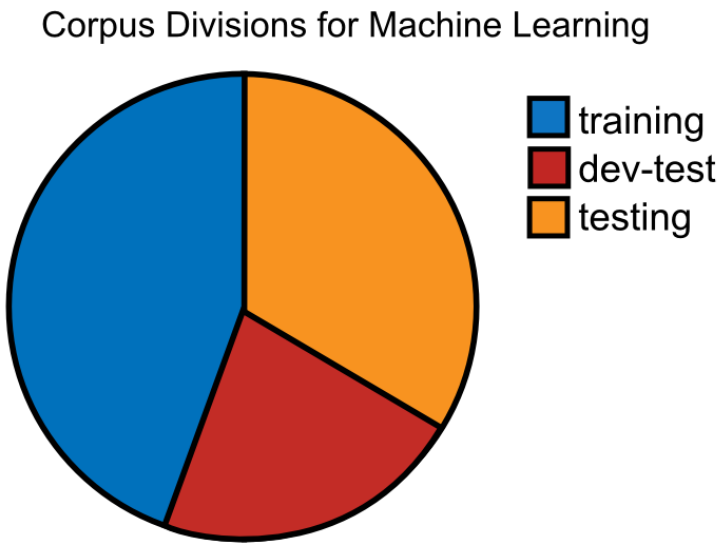


Figure 2.1 – Corpus Divisions for machine learning (extracted from [40])

The training set is used to train the algorithm that is executed over the corpus. The development-test set is used for error analysis, since the set is not "biased" by certain features of the training corpus [20]. Thus, one can obtain an indication on where the algorithm is wrong by tagging the corpus, and carrying out the necessary adjustments to new training and testing rounds if needed.

The test set should be as large as possible, for a small set can be unrepresentative. However, it is also necessary to save the maximum possible data for the training set [20].

Data distribution between sets should keep a significant proportion so one may perform statistical analysis on this data. In practical terms, it is usual to divide data into 80% for training, 10% for development and testing and 10% for final testing [20].

2.2.2 Corpora of Portuguese Texts

In this section, we briefly present some projects that contributed to the creation of corpora of Portuguese texts, placing greater emphasis on Brazilian Portuguese corpora.

One of the most prominent projects in NLP has been Linguateca[4], which maintains a page concentrating studies focused on processing texts in Portuguese. Linguateca offers a series of raw and annotated corpora using the parser Palavras, such as COMPARA, which has texts in Portuguese and their English translations and vice versa [52]. Also noteworthy, CETENFolha (Corpus de Extratos de textos Eletrônicos NILC/Folha de São Paulo) is a corpus of about 24 million words with texts taken from the newspaper Folha de São Paulo and compiled by Núcleo Interinstitucional de Linguística Computacional de São Carlos (NILC).

NILC is also responsible for creating Lácio-Web[5], a set of corpora with 10 million words, composed of texts of contemporary Brazilian Portuguese, created in 2002 in a partnership with Instituto de Ciências Matemáticas e de Computação and Faculdade de Filosofia, Letras e Ciências Humanas of the University of São Paulo. This project also provides linguistic and computational tools (such as POS taggers), including features for researchers from all areas involved.

HAREM[6] (Avaliação de Sistemas de Reconhecimento de Entidades Mencionadas), in its two editions, proved to be very important in the evaluation of the work on Portuguese corpora. To support this event, it was prepared a "Gold Standard", *i.e.*, a manually annotated collection of varied texts from different genres. With this collection, the systems' skills were evaluated according to identification, semantic and morphological classification of named entities [43, 44].

2.2.3 Short Message Corpora Characteristics

SMS is a service used for mass communication. Due to the limited size (160 characters per message) and the users' profile (the majority of the users are teenagers [34]), the language in these messages shows quite peculiar characteristics. SMS texts are highly in-

[4] http://www.linguateca.pt
[5] http://www.nilc.icmc.usp.br/lacioweb/index.htm
[6] http://www.linguateca.pt/harem/

formal, as the communication is based on rapid exchanges of information. According to [34], the main characteristics of SMS language are:

- Phonetic abbreviations, such as "txt" for text, basically removing vowels and replacing consonants by phonetic equivalents;
- Non-phonetic text abbreviations, such as emojis, are also widely used;
- Whitespaces are omitted in order to save characters;
- Some orthographic rules are ignored, such as accentuation rules.

Tweets – which are limited to 140 characters – work similarly to SMS in information extraction tasks. Melero *et al.*, in [30], consider some features and deviations of the language in tweets as challenges to language processing, such as:

- Twitterers emphasize the text with capital letters, featuring words with intensity or significance. For instance, "NÃO é brincadeira" (It is NOT a joke). However, proper names are not always capitalized;
- As with SMS messages, accent marks are frequently omitted;
- Punctuation marks are omitted or repeated, when one wants to emphasize a sentence;
- Informal writing style is found: word abbreviations, such as "pq" for "porque" (because) are intentionally used;
- Spelling and agreement errors are found.

2.3 Corpus Preprocessing and Annotation

Although raw texts can provide a fair deal of information, there are many advantages in structuring corpora, as seen in previous sections. One can delimit the boundaries of words and phrases or, in a more detailed structure, syntactically classify the components of sentences [27].

Tagging or annotating a corpus consists of inserting codes in the text, informing something about the type, structure or meaning, and format of this text.

In [31], Gary Miner *et al.* depict the stages of the NLP task, starting from a textual document to arrive at its meaning. Following the stages shown in Figure 2.2, tokenization is the task of identifying and splitting the input text into lexical items or other tokens. A token can be a word, a number, or even a punctuation mark. Lexical analysis, along with morphological analysis, assign tags to terms and deal with disambiguation of words [31].

Syntactic Analysis is used to determine if the input text is a sentence in the given natural language and describe its syntactic structure [10]. Finally, semantic analysis comprises detecting relations between terms.

Another question concerning corpus preprocessing is regarding to maintain or not word variations, forms or combinations, as in “to sing”, “sing” and “singing”. Stemming comprises working only with the ‘stem’ of the word, *i.e.*, when all affixes (prefixes and suffixes) in a word are removed [27].

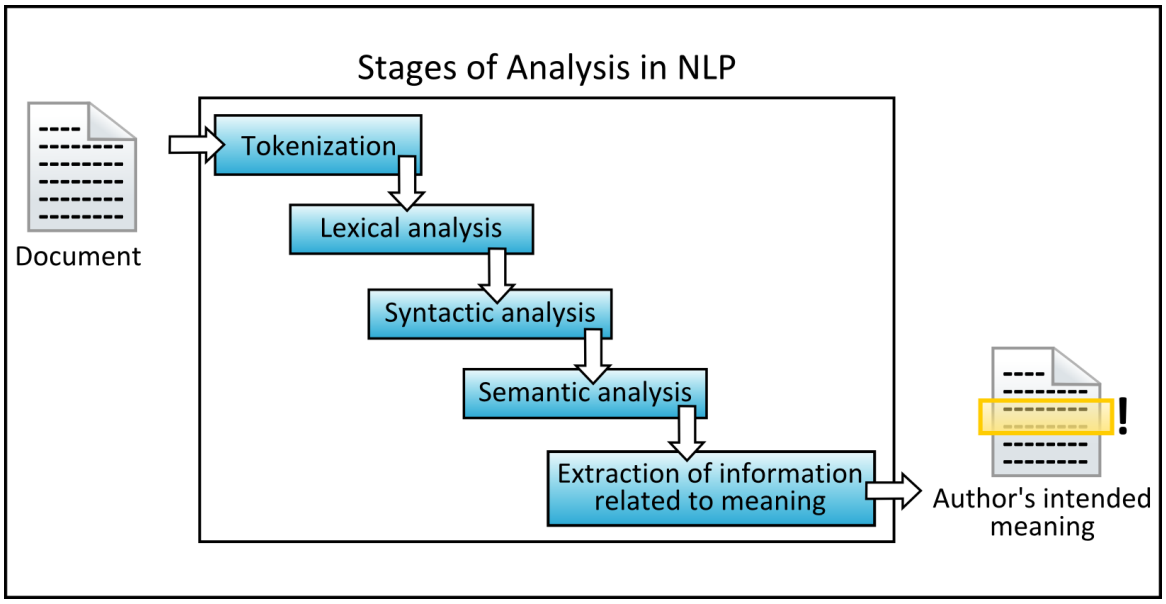


Figure 2.2 – Analysis Stages in NLP (adapted from [31])

2.4 Evaluation

To evaluate IE systems, the most common alternative is to create manually annotated documents. According to [19], “for domain-specific information extraction systems, the annotated documents have to come from the target domain”. Recall, precision and F1 measure are some of the metrics used to evaluate IE systems. Precision, as exposed in Equation 2.1, is the ratio of correct instances (true positives, *tp*) among the identified positive instances (true positives and false positives, *fp*).

$$P = \frac{tp}{(tp + fp)} \tag{2.1}$$

Equation 2.2 shows recall, the ratio of correctly identified instances among all positive instances, according to a manually annotated gold standard (true positives and false negatives, *fn*).

$$R = \frac{tp}{(tp + fn)} \tag{2.2}$$

F-measure provides a way to combine Recall and Precision into a single metric. F-measure is called F1 when precision and recall are equally balanced, as in Equation 2.3.

$$F1 = 2 \times \frac{P \times R}{(P + R)} \tag{2.3}$$

2.5 Chapter Summary

This chapter presented a review on the literature on topics of Information Extraction and Corpus building and processing. Our object of study is a set of SMS messages we should prepare as an input corpus to extract information. These studies led us to propose a corpus building process in Section 4.2, which we executed to build BraCorpSMS, as detailed in Section 5.1.

In this chapter, we also studied the Information Extraction process, which aims to structure data in order to obtain strategic information. Information Extraction comprises identifying names mentioned in texts, temporal information and events of interest. Named Entity Recognition systems generally search for person names, organizations and places. To address that matter, many systems approach NER from a set of hand-coded rules, which demand aid from domain-knowledge experts, but are valuable when in absence of a training corpus. Temporal Expression Recognition is responsible for finding groups of words that correspond to temporal expressions in a text and standardizing them into a computer-processable format. This task can be performed by recognizing lexical triggers - which can be nouns, proper nouns, adjectives or adverbs - and marking up the temporal expression boundaries according to tagging standards, such as TIMEX. The Event Detection and Classification task comprises recognizing events of interest that are mentioned in texts. The most common event detection techniques include rule-based approaches and statistical learning approaches [2]. Techniques used in NER, such as part-of-speech annotation, have also been applied to event detection [20].

As we want to extract information from SMS messages sent in emergency situations, this research will focus on detecting events and relating them to the time they occurred.

3. RELATED WORK

In this Section, we focus on researches involving SMS corpora building, information extraction from short messages, and information fusion. We review related works, commenting on contributions, advantages and putting emphasis on researches most similar to ours.

3.1 Related work on Corpora of Natural Language Short Messages

In this section, we focus on the decisions, methods and steps towards SMS corpus building. SMS corpora studies may have different foci and approaches. Chen *et al.* in [6] point out the scarcity of publicly available SMS corpora, being the majority of them in English. Treurniet *et al.* in [55] indicate that existing SMS corpora differ in size, language and collection method. Next, we detail researches considered to define the SMS corpus building process. Other studies partially aim at corpora related research, and may be of interest in a broader perspective of research [37, 29].

3.1.1 A Corpus Linguistics Study of SMS Messages

In [51], Tagg presents an overview of the literature on text messaging and corpus building in this area. This research is relevant to us as it presents in detail the steps taken to build a SMS corpus. We considered these steps while building the SMS Corpus necessary for our study case (see Chapter 5). Nevertheless, as this research focuses on building a corpus for linguistic studies, it lacks a broader view regarding its usefulness for information extraction purposes.

In order to explain the corpus building process, Tagg introduces CorTxt, a corpus containing over 11,000 text messages. CorTxt contains messages forwarded from friends and family and from a discontinued anonymous online public forum as well. CorTxt is stored as a text-only file (.txt) and as a Microsoft Access database.

To ensure transparency and to protect the privacy of those potentially involved in these messages, Tagg obtained the consent of participants through a form and anonymized text messages. Whilst anonymization cannot remove all personal information without altering data significantly, it must protect the participants' information whenever possible. Tagg chose between a formal and a 'true, complete' method of anonymization. Formal methods anonymize messages automatically, as in [55, 8]. On the other hand, a 'true, complete' method of anonymization is a qualitative method, demanding further analysis, at the risk of

being time-consuming. Therefore, Tagg conducted anonymization semi-automatically, replacing personal names by NAME and a number, surnames by SURNAME, and so on.

This approach was only possible because of Tagg's familiarity with the participants. Tagg decided not to normalize short messages based on the suggestion that the unconventional spellings within these messages create an identity and the normalization process would remove part of texter's intended meaning.

3.1.2 The NUS SMS Corpus

Chen and Kan, in [6], describe the process of building NUS SMS Corpus, a live and public corpus of SMS messages. With more than 71,000 messages in English and Mandarin Chinese, the NUS SMS Corpus is topic independent and is available online[1] comprising XML and SQL dumps along with corpus statistics. This work is significant for us, as it presents a wide variety of methods and exposes the challenges faced while collecting and preparing a SMS corpus.

Chen and Kan propose three methods to collect SMS messages. The first one involves the transcription of messages from a mobile phone. Approaches vary from simply writing down short messages, collecting SMS messages via web form or even taking pictures of messages. The authors point out that, from a large personal network, it may seem easier to collect a fair deal of messages, but results may be biased, lacking representativeness [6]. The second collecting proposed method starts from the establishment of a communication channel where volunteers may contribute sending SMS messages directly to the researchers. The main advantage of this method is that collected messages remain intact. It can happen through phone widgets or specific applications like Microsoft's My Phone[2] [6]. Finally, the third method suggests building a large-scale corpus by extracting SMS messages directly from devices. For this method, it is necessary that contributors forward messages to a specific collection number, resulting in monetary costs. The latter two methods tend to minimize transcription errors and collect messages in an unbiased manner.

To make the corpus publicly available, it was necessary to anonymize identifiers and encrypt contributors' metadata. According to Chen and Kan, using a static release of the corpus may facilitate anonymization as opposed to using a live corpus, where data collection occurs periodically. To address that matter, they had to appeal to the crowdsourcing platforms Amazon Mechanical Turk[3] and ShortTask[4] for English and Zhubajie[5] for Mandarin

[1] http://wing.comp.nus.edu.sg:8080/SMSCorpus/
[2] http://myphone.microsoft.com/
[3] https://www.mturk.com/
[4] http://www.shorttask.com/
[5] http://zhubajie.com

Chinese. During these steps, the authors collected a high amount of data, but a fair part of them contained fake messages, jokes and transcription errors.

We highlight this collecting method as it automatically anonymizes SMS messages on the client application, *i.e.*, before sending them to the corpus. Among others, Chen and Kan replaced email addresses, date and time by, respectively, the strings <EMAIL>, <DATE>, <TIME>. The authors also used regular expressions to anonymize messages, removing numbers with more than one digit.

3.1.3 The Dutch SMS Corpus

Treurniet *et al.*, in [55], present SoNaR SMS corpus, a freely available corpus containing 53,000 text messages in Dutch. The authors collected messages through voluntary donations using a modified version of the NUS Android application. Collected metadata comprised age, gender, place, and country of residence, among others. Treurniet and co-authors were also concerned with anonymizing private information without altering the original text. A unique and encrypted identifier replaced phone numbers. Anonymization codes manually replaced other sensitive data, such as (DATE) for dates, (E-MAIL) for e-mail addresses, etc. This project also involves a website with instructions to donate text messages via e-mail or to submit them online. The corpus was FoLiA XML[6] formatted and tokenized with UCTO[7].

As indicated by Treurniet *et al.*, necessary metadata depend on the object of study. Sociolinguistic studies may need background information on the authors of each message (age, gender, origin, etc.), as behavioral or diachronic studies may demand metadata such as message's delivery date, in order to understand SMS communication along different moments. On the other hand, linguistic studies may require an exact transcription of the original text, with its errors, emoticons and abbreviations.

Besides, Treurniet *et al.* indicate three methods for data collection. The first one being collecting text messages from close contacts, as in Tagg's CorTxt [51]. The second method involves establishing a communication channel so texters can submit the SMSs directly to it. A third method concerns extracting messages from the devices.

[6] http://ilk.uvt.nl/folia
[7] http://ilk.uvt.nl/ucto

3.1.4 The 88milSMS Corpus

Panckhurst *et al.* in [35] present a large corpus of anonymous SMS written in French. The corpus is available on the grid of Huma-Num Services[8] and is part of the sud4science project[9]. The goal of this project is to organize anonymized SMS written in French in a global database, collected from various countries during a period of a decade. This recent research exposes the significance of the subject and how important it is to build such resources for linguistic and natural language processing studies.

The corpus contains 93,085 messages collected during a 13-week period. The authors automatically anonymized messages using the Seek & Hide [1] software, which has an automatic and a semi-automatic phase. Next, messages are transcoded in standard French to receive any subsequent linguistic treatment, in order to restore the spelling and grammar. Finally, an optional annotation phase involved eight types of tags such as grammar, spelling, language, etc. The corpus is organized in a database publicly accessible and is currently used for linguistics, computational linguistics and computer science studies, such as in [26], which proposes a method for aligning text messages to automatically build a SMS dictionary of words in French.

3.2 Related Work on Short Message Processing Architectures

In this section, we discuss related work regarding architectures for processing short messages. These researches present resources, components and techniques relevant to address features of the short message language, which are important to consider for our proposal of a SMS Information Extraction Architecture. Some studies concerning Information Extraction must be of interest for processing texts in standard languages [24, 37, 53, 29], as well as other studies regarding short message language processing [21, 28, 39, 36].

3.2.1 SMS Normalization System

Oliva *et al.*, in [34], present a SMS normalization system to deal with linguistic irregularities and mannerisms, translating SMS messages in Spanish to the standard language. As Kobus *et al.*, in [23], the authors consider word abbreviations as highly phonetically driven, approaching SMS normalization as a machine translation problem. Bearing in mind that there was no Spanish SMS corpus available at the time, they built a corpus

[8] http://88milsms.huma-num.fr/
[9] http://www.sud4science.org/

with 92 SMS messages collected from voluntary contributions. For testing purposes, they built a second corpus translating a Spanish book to SMS language. The proposed system is adaptable for other languages. Regarding the system architecture, the authors developed a system composed of three modules: Preprocessing, translation and disambiguation. The preprocessing module tokenizes the messages, including tokens that have both letters and numbers. The module also upper-cases words at the beginning of sentences. Subsequently, the module tries to translate tokens using a dictionary of Spanish, storing their possible translations. Otherwise, the module separates letters and symbols in new tokens.

As previously mentioned, to accomplish SMS normalization, it is imperative to address SMS features. It is necessary to process special symbols and phonetic abbreviations whilst disambiguating words. Hence, the authors divided the system into three modules, namely preprocessing, translation and disambiguation. The preprocessing module handles messages, aiming a proper tokenization of the text, including letters and numbers, and capitalizing words and sentences. Subsequently, the module tries to translate tokens using a Spanish dictionary, considering their possible translations. Otherwise, the module separates letters and symbols in new tokens. The translation module must find all possible word translations. To deal with out of vocabulary words and non-phonetic abbreviations, the module uses a SMS dictionary with over 11,000 entries, provided by the Asociación Española de Usuarios de Internet and a phonetic Spanish dictionary built especially to handle phonetic abbreviations and real words [34]. The disambiguation module's purpose is to choose the most appropriate translation for each situation. To accomplish this task, the module calculates the similarity of each pair of possible translations preceded by a shallow parsing, which assesses the most likely sequence of POS tags, eliminating some options while avoiding dealing with words with a very high ambiguity level.

For testing, the authors used two data sets: one containing 92 messages written by college students; other being a larger set, consisting of extracted messages from a book in Spanish. The latter was used to check whether the conclusions drawn in the former were usable in other areas and other writing styles. The first data set produced word error rate of 19.5% and 61% sentence error rate, and a BLEU[10] score of about 80% with trigrams. These results are superior to those of other SMS normalization systems [34]. The second data set showed good overall and individual performance.

The method developed by Oliva *et al.* is one of the firsts for Spanish. Despite being focused on this language, it is important to highlight its adaptability to other languages, as it does not rely on large annotated corpora. In addition, one of the major contributions of this study was to present a new similarity measure based on phonetics, which one can use in other researches on word or SMS phonetic processing.

[10]Bilingual Evaluation Understudy is a known metric used to compare machine translations to their actual translation [34].

3.2.2 Information Extraction from tweets during crisis events

In [8], Corvey *et al.* introduce a system that incorporates linguistic and behavioral annotation on tweets during crisis events to capture information about situation awareness. The system filters relevant and tactical information intending to help the affected population. Corvey *et al.* collected data during five disaster events and created datasets for manual annotation. This research interests us, as it involves an architecture for processing a tweet corpus to support information extraction, creating categories of events and classifying tweets according to them.

The authors linguistically annotated the corpus, following the Automatic Content Extraction[11] (ACE) guidelines, looking for named entities of four types: person, name, organization and facilities.

A second level of behavioral annotation assesses how members of the community tweet during crisis events. Tweets receive different and non-exclusive qualitative tags, according to the type of information provided. As a result, tweets containing situational awareness information are collected and go through other annotation steps, being tagged with macro-level (environmental – social, physical or structural) and micro-level (regarding damage, status, weather, etc.) information.

Corvey *et al.* trained a machine learning classifier with about 2,000 tagged tweets containing situational awareness information. They normalized tweets by removing specific symbols, such as 'RT' and '#'. Next, they tokenized and applied part-of-speech tagging and word frequency count. According to the authors, the results indicated that, under emergencies, "users communicate via Twitter in a very specific way to convey information" [8]. Becoming aware of such behavior has helped improving the classifier's performance to an accuracy of over 83% using POS tags and bag of words. To classify location, they used as features Conditional Random Fields (CRFs) with lexical and syntactic information and part-of-speech. The annotated corpus was divided into 60% for training and 40% for testing. The authors obtained an accuracy of 69% for the complete match and 86% for the partial match and recall of 63% for the complete match and 79% for the partial match.

3.2.3 A Lexicon of Verbs for Twitter

Williams and Katz, in [59], describe the creation of a new lexicon[12] containing 486 verbs annotated with the duration of the events they refer to. This project was carried out with a corpus of more than 14 million tweets, supplying what was until then a nonexistent

[11] http://www.itl.nist.gov/iad/894.01/tests/ace/
[12] https://sites.google.com/site/relinguistics/

resource for the English language processing, especially when time information are absent in the text. Through this analysis, one can find the average duration of various types of events, which allowed the authors to infer the duration of events in other tweets, when this information is implied. This research is important for us as it details steps from data collection to processing of tweets and provides insights about how users communicate through social media and the relation between temporal information and events.

The lexicon creation involved collecting, filtering and normalizing data available on Twitter. The authors used POS tags and assigned a unique identifier for each tweet. In line with the preprocessing steps shown in [8], links, references to other twitterers and hashtags were removed. Williams and Katz also tokenized words considering blank spaces. The use of regular expressions was important to recognize sequences. To associate temporal durations to their corresponding events, the authors created four types of regular expression extraction frames: verb + 'for' + duration, verb + 'in' + duration', 'spend' + duration +verb, and 'takes' + duration + verb. From this frames, the authors could extract features, such as verb lemma, tense, duration and type.

The extracted corpus contains approximately 400,000 tweets, containing lemmas of 486 verbs. Precision was measured in 400 randomly selected tweets, while the authors checked manually if it was possible to extract the verb, the verb tense, expression and duration. After evaluating the extraction frames, Williams and Katz obtained overall precision of 90.25%. They also trained a machine learning classifier to identify tweets according to their habituality (habit or episode). The authors hand-labeled a random sample of 1000 tweets, and the classifier obtained an accuracy of 83.6% during training. The extracted corpus was classified into 94,643 tweets mentioning habits and 295,918 tweets mentioning episodic events.

With these results, one can focus on automatic temporal interpretation problems. For example, with this lexicon, one can distinguish the use of terms like "shortly after", which can indicate either an immediate action, or an action that took a slightly longer period of time.

3.2.4 A Framework for Translating SMS Messages

Sridhar *et al.*, in [48], present an application of statistical machine translation to SMS messages. According to the authors, "translating SMS messages has several challenges ranging from the procurement of data in this domain to dealing with noisy text (abbreviations, spelling errors, lack of punctuation, etc.) that is typically detrimental to translation quality" [48]. This research is relevant for us because it details the process of data collection and the steps and resources used on the SMS message translation framework.

The Framework uses finite state transducers to learn the mapping between short texts and canonical form. Due to the absence of training SMS data, the authors used a corpus of tweets as surrogate data, which they consider a good approximation to real SMS data. According to Sridhar *et al.*, "the language used in Twitter is similar to SMS and contains plenty of shorthands and spelling errors even though it is typically not directed towards another individual" [48]. They built a bitext corpus from 40,000 English and 10,000 Spanish SMS messages, collected from transcriptions of speech-based messages sent through a smartphone application. Another 1,000 messages were collected from the Amazon Mechanical Turk. 10,000 tweets were collected and normalized by removing stopwords, advertisements and web addresses.

Furthermore, the framework processes messages in Spanish (lowercased and unaccented), segmented into chunks using an automatic scoring classifier. Abbreviations are expanded using expansion dictionaries (built semi-automatically) and then go through a translation model based on sentences. The authors used an unsupervised approach to learn the normalization lexicon of word forms used in SMS messages. From this, they built a static table to expand abbreviations found in SMS messages, where a series of noisy texts have the corresponding canonical form mapped. For example, the noisy form "4ever" is linked to the canonical form "forever". Next, the framework comprises a phrase segmentation component that uses an automatic punctuation classifier trained over punctuated SMS messages. Finally, the Machine Translation component consists of a hybrid translation approach using a phrase-based translation framework with sentences from the input corpus represented as a finite-state transducer. The framework was evaluated over a set of 456 messages collected in a real SMS interaction, obtaining a BLEU score of 31.25 for English-Spanish translations and 37.19 for Spanish-English.

3.3 Related Work on Information Extraction Applications

In this section, we will present researches related to different applications of IE systems. We start by briefly discussing researches that pay special attention to display the extracted information. Next, we present researches similar to ours in more details.

Dias and Fonseca, in [12], introduce MuVis, an interactive visualization tool for large music collections, based on music content and metadata. The tool combines a user-centered design with information visualization techniques, music information retrieval mechanisms (for semantic and content-based information extraction) and dynamic queries, to browse for music collections and to create playlists. Moreover, the tool provides similarity mechanisms to listen to related music. In order to do that, the authors conducted an online survey and a contextual inquiry to discover how users find and explore music, and how they generate playlists. To achieve these purposes, the solution has an architecture that consists

of: a feature extraction module, responsible for extracting common tags (genre, track title, duration) and fluctuation patterns; a retrieval component that looks for similar tracks according to their tags or their content information; the user interface, which includes a list view and a spatially ordered treemap. The treemap view presents items by their similarity to the main artist, which can be changed anytime, allowing users to reorganize it. The properties of each node, like size and color are customizable.

In [33], Ning *et al.* present OncoViz, a tool for text mining and visualization for health related documents. The main contributions of this work are to propose methods to discover relevant information from a large collection of text documents in form of associations and visualization of extracted knowledge in an integrated view of multiple drugs and their side effects. Text mining helps providing newly reported data as it can automatically retrieve and mine documents and integrate the results with the visualization tool. To show only pertinent information, the tool allows users to select drugs from a list and "zoom in" to see their side-effects.

In [57], Wang *et al.* introduce Timely Yago, a prototype to extract temporal facts from Wikipedia[13] texts and integrate them into its knowledge base. Timely Yago's ontology uses a RDF-style (Resource Description Framework) model, where all objects are represented as entities and an instance of a relation between two entities is called a fact. The authors have created the concept of temporal facts to associate time information to the entities. Temporal facts may be valid during a specific time or within an interval. The authors use regular expression matching to detect the validity time of facts on the semi-structured elements in Wikipedia. Rule-based techniques can also trigger learning-based methods applied on the text. In this work, events are defined as tensed verbs, adjectives and noun phrases that detail temporal information of the events.

The prototype also supports SPARQL-style (SPARQL Protocol and RDF Query Language) language and displays the results in a timeline, based on SIMILE Timeline[14]. For validation purposes, the prototype contains around 300,000 temporal facts from the sports domain where a series of temporal queries are supported, such as checking data on a certain player or who are his teammates.

In [47], Spasić *et al.* describe a research on automatic information extraction from reports containing medications used by patients. For this extraction, the authors took into account the drug name information (m), dosage (do), mode (mo), frequency (f), duration (du) and reason (r) of the medical discharge summaries. The records considered were structured, following a model with predefined spaces, to contain only relevant information. The sentence "In the past two months, she had been taking Ativan of 3-4 mg q.d. for anxiety" would contain the corresponding model: m="ativan", do="3–4 mg", mo="nm", f="q.d.", du="two months", r="for anxiety".

[13] http://www.wikipedia.org/
[14] http://www.simile-widgets.org/timeline/

3.3.1 Open Domain Event Extraction from Twitter

Ritter *et al.*, in [41], present TwiCAL, an open-domain event extraction and categorization system for Twitter. This research is relevant for us, as it proposes a process for recognizing temporal information, detecting events from a corpus of short messages and outputting the extracted information in a calendar containing all significant events among them. As the authors affirm, “The short and self-contained nature of tweets means they have very simple discourse and pragmatic structure, issues which still challenge state of the art NLP systems”.

Ritter *et al.* used a corpus of tweets annotated with events as training data for sequence labeling models. In order to associate an event to a calendar zone, they focused on identifying events referring to a unique date.

TwiCAL extracts a 4-tuple representation of events, including a named entity, an event phrase – consisting of many different parts of speech, a calendar date, and an event type. The architecture, whose input is a set of raw tweets, comprises six components: POS Tagger, NER, Temporal Resolution, Event Tagger, Event Classification, and a Significance Ranking.

Ritter *et al.* trained a POS tagger and a named entity tagger on in-domain Twitter data. To build an event tagger, they trained sequence models with a corpus of annotated tweets.

To resolve temporal expressions, the authors used TempEx, a rule-based system that uses a time anchor and POS to mark temporal expressions on a text. Noisy temporal expressions remained unhandled.

The open-domain event categorization uses variable models to discover types that match the data and discards any incoherent types. The result is applied to the categorization of extracted events.

The classification model is evaluated according to the event types created from a manual inspection of the corpus. The authors compared the results with a supervised Maximum Entropy baseline, over a set of 500 annotated events using 10-fold cross validation. Results achieved a 14% increase in maximum F1 score over the supervised baseline. A demonstration of the system is available at the Status Calendar webpage[15].

[15] http://statuscalendar.com

3.3.2 Detecting Competitive Intelligence from Social Media

In [11], Dai *et al.* present SoMEST (Social Media Event Sentiment Timeline), a framework for Competitive Intelligence analysis for Social Media and the architecture of a NLP tool combining Named Entity Recognition, Event Detection and Sentiment Analysis. Competitive Intelligence (CI) is the process of gathering information about competitors and the competitive environment to plan processes and decisions to gain competitive advantage [11]. This research is similar to ours, as it presents an architecture to extract information from social media texts and the visualization of this information.

The authors use Event Timeline Analysis (ETA) to detect events and display them in a timeline, highlighting trends or behaviors of competitors, consumers, partners and suppliers. From the visualization of this information, a corporation may analyze events caused by its competitors and prepare a preemptive strategy. Dai *et al.* also use Sentiment Analysis to measure human opinions from texts written in natural language, searching for the topic, who expressed the opinion and if it is positive or negative.

The process defined by the authors comprises three phases: data collection, extraction and classification, and synthesis. Starting from collecting social media texts generated by customers, SoMEST focus on detecting events published from companies and opinions shared by the customers. The extraction and classification phase consists of analyzing data and generating event extracts and opinion extracts. These extracts are synthesized into a social media profile that unifies events and opinions linked to the leaders, brands, services and products of a corporation into a period of time. Event Detection and Named Entity Recognition are used to extract information like the event, its actor of the event or the topic of sentiments. Finally, the timeline displays a chronological order of the corporation's events, the competitors' events and changes in customers' opinions. The authors also demonstrate a practical example of SoMEST, analyzing tweets to extract sentiment information on two brands of Tablet computers. Tweets were collected using the Twitter Sentiment tool[16].

3.4 Chapter Summary

In this chapter, we reviewed related work regarding short message corpora, information extraction architectures, and applications for the extracted information. Related work reveals common steps that let us propose a process which fulfills the needs for building SMS corpora turned to information extraction facing crisis events. Furthermore, we could learn from their difficulties while gathering, preparing and formatting data. We detail our choices and recommendations in Chapter 4.

[16] http://twittersentiment.appspot.com/

Accordingly, as we can observe by analyzing researches in Section 3.2 and Section 3.3, even IE systems built for different tasks may present similarities. In light of this, we could understand common points in different IE architectures, mainly due to the nature of short text messages. From this learning, we could propose an architecture for Information Extraction from SMS messages according to core components shared by most IE systems reviewed in this chapter, such as POS taggers, tokenization, and normalization, while adding other components to treat domain-specific characteristics.

Other researches were not directly related to ours, but helped us understand the context of our work. For instance, in [14], Gonzalez presents a study on how Portuguese is used on the internet. This research is relevant for us, since this is the same language used in SMS messages and tweets. The research explored a corpus of about 135,000 words collected from blogs with the purpose of analyzing the most frequent words and their variation frequency. Through corpus analysis, it was possible to assess features of the language on the internet such as the use of abbreviations, substitution of letters and accents and keystrokes. Another important knowledge source worth mentioning is the book "Mining the Social Web", by Matthew Russel [42], which even though turned to social networks served us as a guideline while prototyping the IE system for the Case Study detailed in Chapter 5.

4. INFORMATION EXTRACTION ARCHITECTURE

During crisis events, short text messages such as SMSs are sources of useful information. These messages contain information that can be extracted, providing valuable resources to support decision-making under these situations. With this in mind, and based on the fundamentals studied in Chapter 2 and related researches reviewed in Chapter 3, in this chapter, we detail the proposal for the IE architecture, which is the major contribution of this dissertation. The architecture addresses Linguistic processing, as well as information extraction tasks such as Temporal and Event Processing along with a component for organizing the extracted information in a human-understandable manner. Moreover, based on other corpora building researches, we propose a process to build a corpus of SMS messages whose input is a set of raw short messages.

4.1 Overview

As seen in Figure 4.1, the proposed IE Architecture takes as input a Corpus of SMS messages in XML format. Then, the Linguistic Processing component is responsible for preprocessing each message and preparing them for Information Extraction. The Temporal Expression Tagger component recognizes and tags all temporal information within the messages, while the event tagger identifies domain-related events and tags them accordingly. The Information Fusion component displays the extracted information so as users of this system can interpret its results. The output of the system is the extracted information organized in a readable display, regarding the application.

Since the input for the IE Architecture is a corpus of SMS messages produced during a crisis event, one needs to build a new input for every new test. In Section 4.2, we propose a simple and straightforward process to address that matter. Next, we detail each component of the IE Architecture in sections 4.3 through 4.6.

4.2 Corpus Building Process

Considering the need for building SMS corpora from messages produced during a certain emergency, and based on other corpora building researches, we propose a process for building a corpus of SMS messages. The input of the corpus building process is a set of raw short messages. The process comprises five steps, namely data collection, data filtering, anonymization, normalization, and formatting and its output is a SMS corpus. Figure 4.2 presents these steps in a workflow. We use, in this figure, a dotted box around the

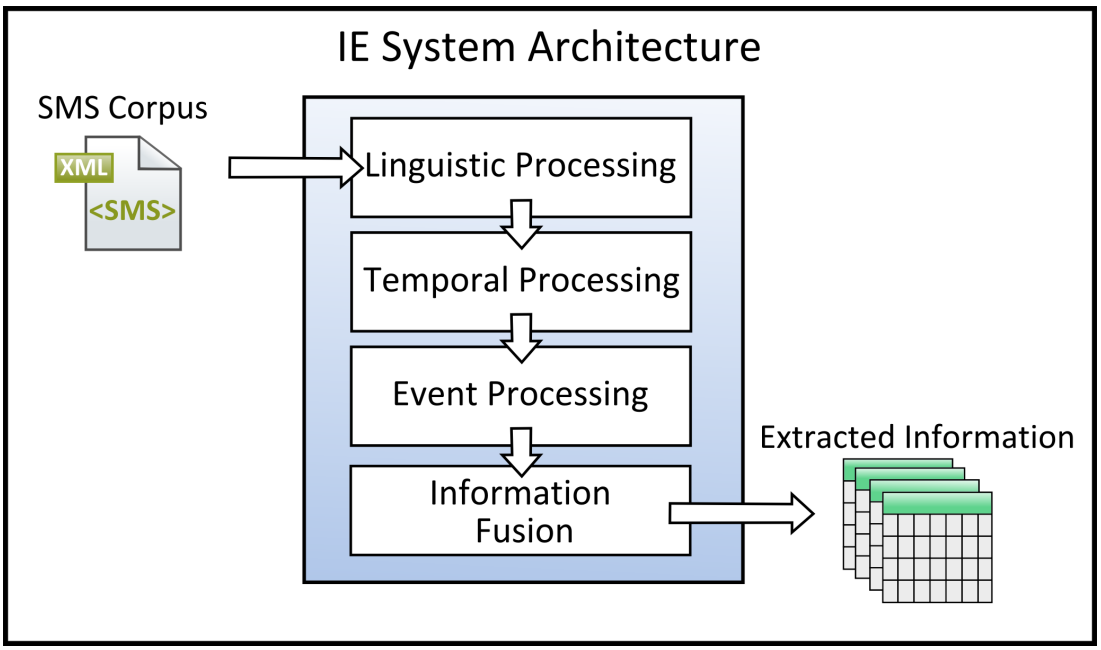


Figure 4.1 – IE Architecture overview

normalization and anonymization steps as they are optional, depending on the purpose of the study and future use of the corpus. We detail each step in sections 4.2.1 through 4.2.5.

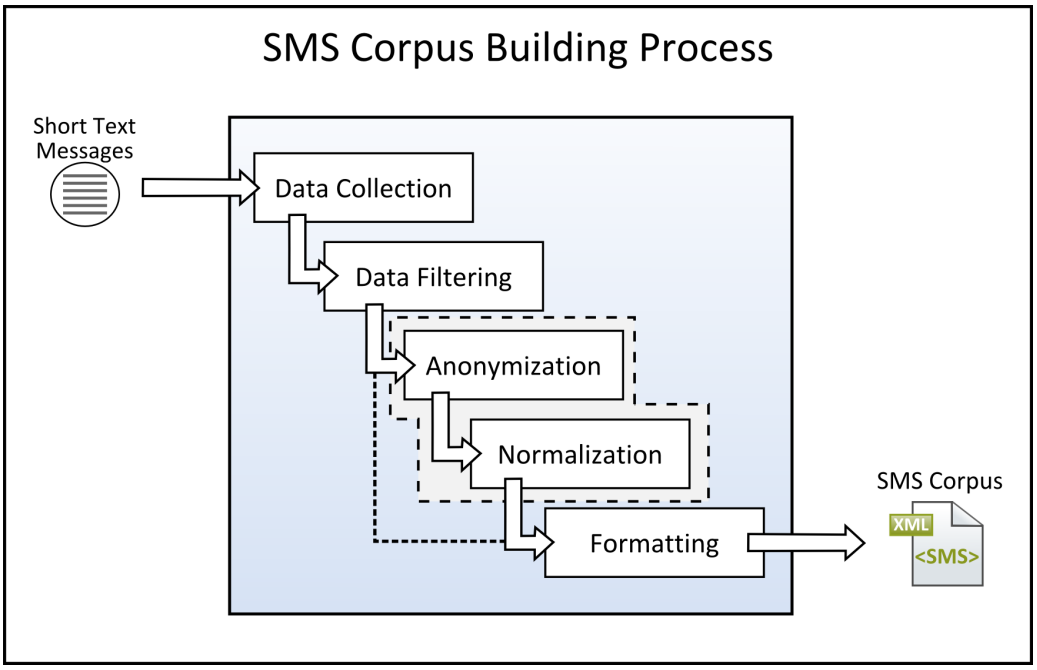


Figure 4.2 – SMS Corpus Building Process

4.2.1 Data Collection

The starting point of the process comprehends collecting SMS messages. According to previously reviewed researches [6, 55], data collection can be performed with one of the following methods: exporting messages via software, transcribing or extracting messages directly from devices, establishing a communication channel or appealing to close contacts.

As building a large-scale corpus depends on involving a large audience, automatic methods like crowdsourcing are more expensive but are recommended as they allow collecting richer and more diverse data. However, it is worth mentioning that attracting the attention of volunteers can be challenging and it is not uncommon to appeal to strategies like raffles, as in [55], or surrogate data, as in [48].

Since information extraction applications have a more specific purpose, especially when compared to linguistic studies, there is the need to choose the more adequate method of data collection for that end. Despite the private nature of SMS messages, when in emergency situations, valuable information is broadcast along communication channels established by service providers, making them a rich datasource.

4.2.2 Data Filtering

Building a corpus of SMS may require collecting as many messages as possible. Nevertheless, not every message may fit the corpus. Thus, it is important to choose an adequate filtering method and to analyze filtered messages to check whether they fit the intended purpose of the corpus. While aiming for information extraction, it is essential to observe which features may indicate good filter candidates. For instance, while looking for temporal expressions in a text, it may be interesting to filter messages containing words like ‘hours’ or ‘minutes’. In some studies, as in [51, 55], contributors can previously select which messages they will send, possibly minimizing the manual work of this step.

Filtering data may be more effective while working with small-scale corpora or when dealing with repetitive messages. Indeed, it is important to have at least a simple filtering level, as to avoid building a corpus with a very broad purpose. However, we recommend defining rules to select representative data whenever possible. Regular expressions may serve this purpose and can be used in almost every existing applications and programming languages.

Furthermore, this step aims to remove inappropriate messages, such as spam and corrupt data. As incomplete or irrelevant messages – and sometimes duplicates - can bias future analysis, one must remove them as well.

To filter unwanted textual data, one can use regular expressions, manual removal with some of the many text editors, like Vim[1], or use spreadsheet applications available, like Excel[2]. Since sifting through a large set of SMS messages in search of valid entries can consume valuable time, we indicate that semi-automatic approaches, such as defining rules to filter out unnecessary data are highly recommended.

4.2.3 Anonymization

Privacy is a concern in almost all research involving short text messages, so it is important to consider anonymization before building the corpus. This may be the case for some applications in information extraction.

As what to anonymize depends on the content of each message, such task is not obligatory in our process. One can approach anonymization manually, by analyzing messages and replacing or removing sensitive information. This method seems to be more applicable for shorter datasets. Another approach consists of dealing with messages programmatically, creating rules to define what are relevant information. There are disadvantages to both methods, since an automatic anonymization approach tends to leave out indirect references and manual anonymization is laborious and depends highly on subjective criteria. As an alternative, we indicate anonymizing messages semi-automatically by creating rules first and then analyzing a minor set of exceptions.

4.2.4 Normalization

Normalization of user-generated content, like SMS data, is a challenge for most current text processing tools, which were not trained to process noisy texts [55]. Short messages tend to contain many lexical variants, such as typos, phonetic substitutions and *ad hoc* abbreviations, influencing the performance of such tools. Han *et al.* in [16] propose a dictionary-based method to normalize lexical variants within SMS messages and tweets. Other researches, like [34, 23, 25] approach normalization as a machine translation problem. On the other hand, Kobus *et al.* in [23] and Weiser *et al.* in [58] indicate normalization is not a necessary step for temporal expression extraction on SMS messages. Tagg also avoids normalizing messages, as it would change their intended meaning, which is important for a linguistic study [51]. Based on these considerations, we indicate normalization as optional.

[1] http://www.vim.org/
[2] http://office.microsoft.com/en-001/excel/

4.2.5 Formatting

In general, text corpora are released in various formats, such as TXT, CSV, XML, HTML, PDF, among others. One must choose the corpus format taking into account the purpose to be achieved. For instance, when aiming at information extraction it is important to export corpora in a compatible format with the most common IE tools. The format decision affects corpus use, as some tools can be incompatible with the format. As, to our knowledge, XML is the format used in most of SMS corpora, it is our recommendation.

4.3 Linguistic Processing Component

The proposed IE Architecture is designed with a component responsible for preparing the messages for that end. This Linguistic Processing Component comprises a Preprocessing module, including four steps: Normalization, Sentence Splitting, Tokenization, and Stopword Removal; and steps specifically designed for linguistic processing, therefore adapted to the language in which the messages are written: POS Tagging and Spell-Checking.

Figure 4.3 shows the Linguistic Processing Component workflow, which preprocesses the SMS Corpus for Information Extraction. We represent the component's external resources as a stack of data while internal resources, such as regular expressions are represented as a circle. The Normalization step is responsible for adjusting the text while facing spelling variations, abbreviations, treating special characters and other features of the short message language. Next, the Sentence Splitting step divides each message into a list of sentences in order to process them individually. Both steps are triggered by regular expressions. Then, in the Tokenization step, each sentence is broken in tokens and Stopword Removal step follows. Every token is compared to a list of stopwords, which enables discarding unnecessary items and speeding up the process of information extraction.

Accordingly, the tokens are tagged with a Part-Of-Speech Tagger, which is trained with an annotated corpus of messages. The following step in the Linguistic Processing Component comprises a Spell Checker, which makes use of an external dictionary to label untagged tokens and submits them to the POS Tagger for revision. As exposed in Figure 4.3, this component outputs a set of preprocessed sentences that serve as input for the Temporal Processing Component.

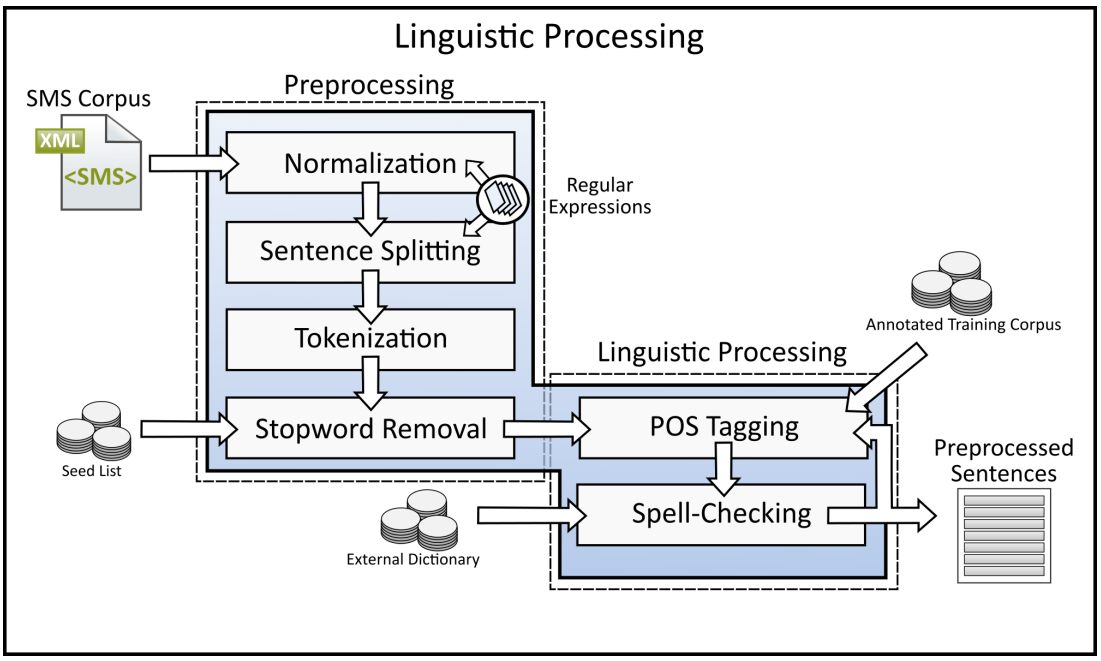


Figure 4.3 – Linguistic Processing Component Overview

4.4 Temporal Processing Component

The Temporal Processing Component is responsible for applying regular expressions in order to identify temporal expressions related to events in SMS messages. Since temporal expressions are limited to a fixed set of syntactic patterns, most Temporal Expression Recognition systems make use of rule-based methods to recognize syntactic chunks [40]. Figure 4.4 shows an overview of this component.

Initially, the Temporal Expression Recognizer uses a rule-based approach to identify variations of temporal references mentioned in the sentences. Even though the rule set is able to identify simple temporal expressions present in messages, rather complex expressions, e.g., "desde às 8h de domingo" (since Sunday 8am), are still to be treated. For these cases, the Temporal Expression Recognizer counts on a list of temporal keywords (in Figure 4.4, see Lexical Triggers) to determine the extent of these temporal expressions.

The Temporal Reference Classifier analyzes the expression according to its lexical triggers and defines the type and value of the temporal expression. Finally, the component tags the temporal expression according to the TIMEX standard (as presented in 2.1.2).

4.5 Event Processing Component

The Event Processing component starts from the Event Detection step, which is responsible for finding relevant events in a sentence. This step counts on a set of rules to

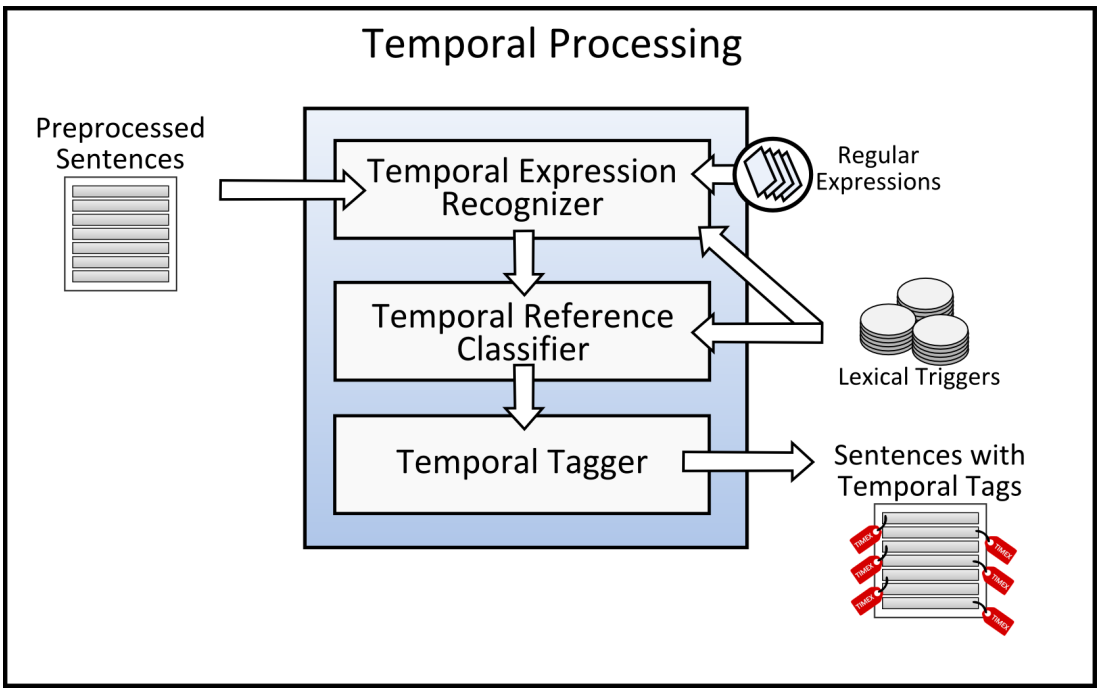


Figure 4.4 – Temporal Processing Component Overview

identify the event, and a "Structure Validator" to check if the structure of the detected event is in a certain format. If so, the event can now be classified.

Since the proposed IE architecture aims to extract information from messages during emergencies situations, one can determine a certain set of categories of events to detect during this step. For instance, as discussed in 3.2.1, Corvey *et al.* in [8] propose a situational awareness annotation level with the intention of understanding crisis events as a whole. To address that matter, the authors define categories such as ‘Social Environment’, ‘Built Environment’ and ‘Physical Environment’. Each category has subcategories with specific information, such as ‘Crime’, ‘Damage’ or ‘Weather’.

Accordingly, in order to be executed, the Event Processing Component requires a previous definition of a set of domain-related observable categories. Consequently, sentences that match any of these categories pass through a classification step, which relates the event to the categories. This step makes use of a list of domain-related keywords. Then, the component can assign the correspondent tags to the event mention. Figure 4.5 presents the overview of the Event Processing Component.

4.6 Information Fusion Component

This component is responsible for grouping and organizing all tagged information in a human understandable manner. All relevant information are “fused” to show the results of the IE application.

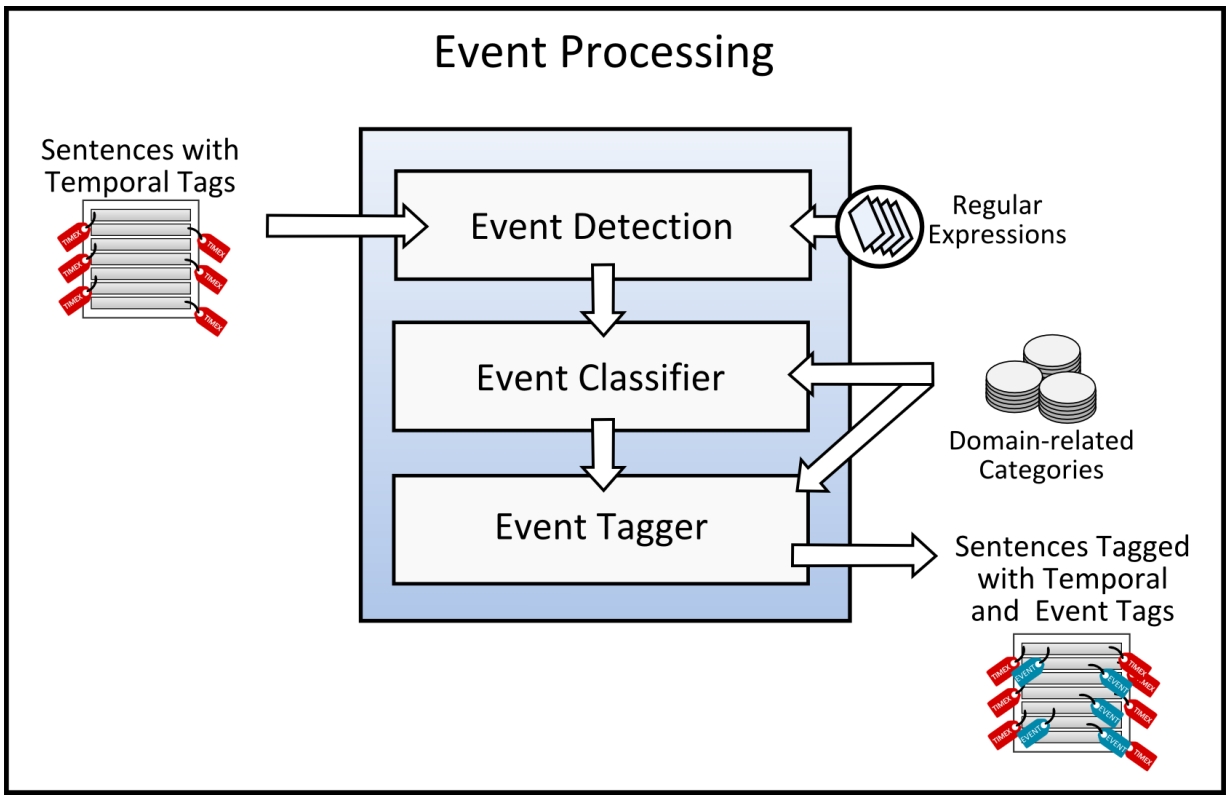


Figure 4.5 – Event Processing Component Overview

There may exist several ways to represent the results, as this decision is linked to the intended purpose of their application. For instance, Ritter *et al.* [41], display their results in form of a calendar, where the respective events are shown. On the other hand, Dai *et al.* [11] present the Information Extraction results in a timeline, showing the progress of event mentions and the amount of associated opinions.

Given a set of tagged events and their corresponding temporal information, the relation between and event and when it occurred must be clearly expressed. An IE system built on this architecture must display the extracted information in a meaningful and relevant way to provide situation awareness and to aid human beings on making their decisions during emergencies.

4.7 Chapter Summary

In this chapter, we presented the proposed IE Architecture for extracting information from SMSs produced during emergencies. The IE application to be based on this architecture will need and input corpus, which is built with messages exchanged during an emergency. This means that new corpora must be built for each new IE application run. For that matter, we also described a SMS corpus building. The corpus building process consists of six straightforward steps that cover from data collection to formatting the corpus.

The IE architecture, in turn, consists of a Linguistic, a Temporal and an Event Processing steps and an Information Fusion component. These components are designed to extract information from the SMS messages of the input corpus and display them in a manner that facilitates observing relevant information.

Moreover, the components should be configured according to the language in which the messages are written and the context of their application. As considering an event relevant is context-related, different IE systems built on this architecture would approach Event Detection differently. For instance, an electric utility company might be interested in observing weather conditions, like storms, or events that affect the power grid. However, a medical care service could focus on observing traffic conditions, riots, fires, and so on.

5. CASE STUDY

In order to validate our proposal, we present a case study that uses this IE Architecture. In this Chapter, we detail our choices, decisions made, experiments and results achieved.

The input data for the process was organized from a set of short messages received by an electric utility company. The clients notify the company when there is a power outage, sending short messages that basically inform the word “LUZ” (light) and the installation number (provided by the company). As observed in messages received, company’s clients use this communication channel to provide situation awareness information, which is currently not yet processed but of great help in services provision. It is important to extract information from these messages to deliver relevant and strategic information about emergencies to restore power to customers as quickly and safely as possible. Examples of information provided in these messages are the time, type and severity of the disaster, obtained from terms found in the text.

5.1 Building BraCorpSMS

As the application of this IE Architecture requires a corpus of SMS messages, we built this corpus from a subset of the messages received by the aforementioned electric utility company. From now on, we will refer to this corpus as BraCorpSMS, which stands for ’Brazilian Corpus of SMS Messages’. In order to build it, we followed the steps previously proposed in Section 4.2. The idea is to provide input to an information extraction system that will then process BraCorpSMS. From Section 5.1.1 to Section 5.1.4, we describe the process of corpus building and its results and comment on the difficulties found.

5.1.1 Data Collection

The collection step makes use of a communication channel where the company’s clients may contribute by directly sending SMS messages (as mentioned in Section 4.2.1). These messages are then stored in an Oracle database. For the sake of the intended use of the information, we decided to collect messages on a calendar basis, considering peak periods from known disaster situations first. The data collection considered messages received during the year 2012. The amount of messages received during this period is presented in Figure 5.1.

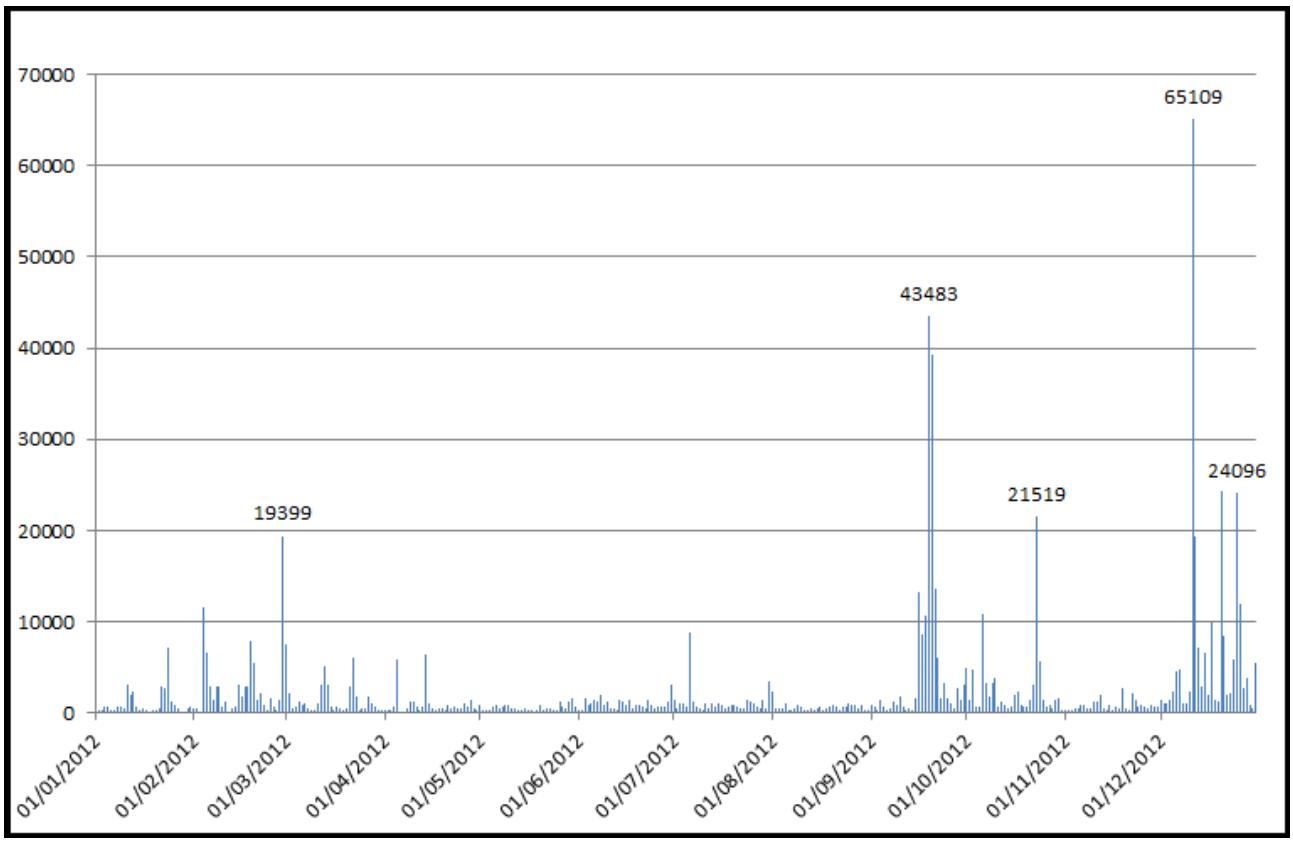


Figure 5.1 – Number of Messages received in 2012 (peak values indicate crisis events)

Table 5.1 shows incoming messages. They are written in Portuguese, capitalized, without accents. Message length varies from 30 to 126 characters in these examples. Each row in Table 5.1 represents a different message and the string #IDNUM was added to replace the consumer unit's identification number.

Table 5.1 – Sample messages (raw corpus)

Message
N.INSTALACAO #IDNUM. EM FRENTE A MINHA CASA DEU UM CLARAO, UM BARULHAO E UM ESTOURO MUITO FORTE NOS FIOS DE LUZ.ESTAMOS SEM LUZ.
ESTOU DOIS DIAS SEM LUZ #IDNUM
LUZ. NUMERO DE INSTALACAO #IDNUM, ESTAMOS SEM LUZ A MAIS DE UMA HORA
LUZ #IDNUM TA PEGANDO FOGO NO POSTE

Typically, during the year 2012, the daily average of SMSs received was around 300 messages. However, crisis events such as storms or fire may cause blackouts and damages, affecting a large area for days. Under these circumstances, the daily amount of messages is likely to increase considerably. The number of messages tends to remain high for more than one day, possibly indicating the disaster elapsed time and the severity of the impact on the affected area. This makes relevant to extract dates and duration of events, by analyzing short messages from a certain time interval.

5.1.2 Data Filtering

From nearly 800,000 messages, we kept only those received during the week between September 16 and September 22 of the selected year. This decision was based on the high amount of messages received throughout this week, which might indicate that mass emergencies occurred during this period. As we want to extract information from messages sent in such a situation, we believe this is a good scenario to learn from. Figure 5.2 shows the amount of messages received during this week, on a daily basis.

Besides, we selected messages with at least 25 characters, considering the minimum expected content for a message is 'LUZ' and the installation number (a 9 digit sequence). Thus, at least other 13 characters would be available to work with. Under these conditions, we gathered over 10,000 messages.

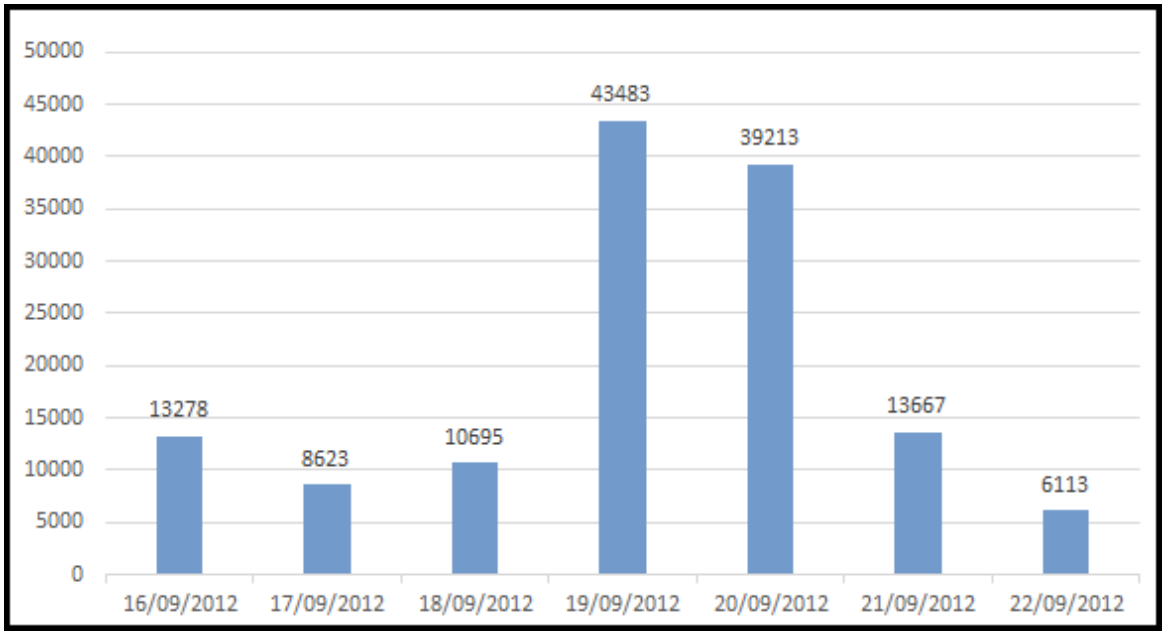


Figure 5.2 – Number of messages received during the week from 16 to 22 September 2012

With a shorter set of messages, we decided to select those containing at least one reference to a temporal expression or an event description. Even though the existing communication channel already discards spam messages, some of them pass along. Therefore, we approached this step by creating simple rules to filter and remove these spam messages, as they all contained references to websites and were sent by a few specific cell phone numbers. We discarded duplicates as well, resulting in a set of 3,021 short messages.

Moreover, we selected only the messages' metadata that are of interest for future information extraction, namely delivery date and message content. This structure may facilitate possible comparisons between temporal expressions extracted from messages and allows us to use the delivery date as a time anchor. Table 5.2 shows the messages' format in three columns after filtering the data: Message ID, Delivery Date and the Message *per se*.

Sensitive information are replaced by #IDNUM at the Anonymization step, which is detailed as follows.

5.1.3 Anonymization and Normalization

To ensure the privacy of texters, we decided to anonymize the identification numbers (installations and house numbers) in the messages. We then replaced them by the tag #IDNUM, as seen in Table 5.2. Since the identification number is a nine digit sequence, we used regular expressions to replace this sequence, preserving other numbers, as we want to keep messages as close to the original as possible.

Even though Weiser *et al.* in [58] indicate that a normalization phase is not necessary to extract temporal expressions from short messages, we decided to address some minor issues, such as removing repeated punctuation marks and blank spaces, as suggested by Kobus *et al.* in [23]. Capital 'O's instead of zeros were very common, so we had to deal with this particularity in our regular expressions too. Digits indicating hours, days and telephone numbers remained intact. As messages were already capitalized and without accents, we opted to keep them this way. Besides, as signaled by Tagg in [51], we aimed at avoiding the risk of changing the messages' intended meaning.

Table 5.2 – Cleaned and Anonymized Messages

Message ID	**Delivery Date**	**Message**
522661	09/20/2012 17:05:25	LUZ #IDNUM TENHO UM IDOSO DE 97 ANOS EM CASA, ESTAMOS MAIS D 24H SEM LUZ, MTO TRANSTORNO. QUERO RETORNO URGENTE.
522667	09/20/2012 17:05:28	LUZ TEM UM FIO CAIDO NO MEIO DA RUA E TEM CRIANCA BRINCANDO VCS VAO ARUMA OU NAO?#IDNUM
522692	09/20/2012 17:06:01	LUZ NUMERO #IDNUM FIO CAIDO DESDE MANHA DE 19-O9
522706	09/20/2012 17:06:21	INST #IDNUM HA 24H SEM LUZ. O QUE FAZER?
522715	09/20/2012 17:06:39	OLA ESTOU SEM LUZ A 24H MORO NA RUA VEREADOR ELBERTO MADRUGA #IDNUM BAIRRO CRUZEIRO PELOTAS

5.1.4 Formatting

We exported the messages from the database in a CSV file, but in order to prepare messages for information extraction, we converted the corpus into a XML format, as this format is recognizable for most information extraction tools and it allows more flexibility while describing the data structure. The XML file (similarly to indicated in [55, 6, 51]) comprises the tag <MENSAGEM> for each message, and as metadata an identification number (ID) and the message's delivery date (DATA). The already anonymized message content receives the tag <TEXTO> and is placed under <MENSAGEM>. Figure 5.3 shows a sample of BraCorpSMS with this tags.

```
- <smsCorpus date="2014.06.28" version="0.1">
  - <MENSAGEM ID="419524" DATA="16/09/2012 00:03:09">
    - <TEXTO>
        LUZ NUMERO DA INSTALACAO #IDNUM AINDA ESTAMOS SEM LUZ
      </TEXTO>
    </MENSAGEM>
  - <MENSAGEM ID="419767" DATA="16/09/2012 07:42:39">
      <TEXTO>LUZ ESTAMOS SEM LUZ DEUS DE ONTEM #IDNUM </TEXTO>
    </MENSAGEM>
  - <MENSAGEM ID="420583" DATA="16/09/2012 09:52:48">
      <TEXTO>LUZ#IDNUM REDE EM MEIA FASE</TEXTO>
    </MENSAGEM>
  - <MENSAGEM ID="420604" DATA="16/09/2012 09:54:57">
      <TEXTO>LUZ.#IDNUM FALTANDO DES DE 4 HORAS DA MANHA.</TEXTO>
```

Figure 5.3 – SMS Corpus Sample with XML tags

5.1.5 Dividing the Corpus

After collecting messages and building BraCorpSMS, we divided the corpus, so to reserve a part for testing and another for prototyping, aiming the representativeness of the data and avoiding a selection bias. Since we decided on following a knowledge-based approach, we need as much data as possible to consult while defining a set of rules, developing features and understanding the SMS language. Based on this decision, we split the corpus in two. The first one, which we defined as a 'Learning Corpus' contains two thirds of the messages in BraCorpSMS; the second, a 'Test Corpus', will be used to assess specific steps of our prototype.

Our tests with the Learning Corpus are followed by an evaluation of the prototype's taggers. With the results of the evaluation, we will use the Test corpus to improve the pro-

totype. Therefore, we divided the Test Corpus in two parts, where the second comprises a set of 100 SMS messages randomly selected (and removed from the initial Test Corpus). Figure 5.4 shows the corpus partitioning. This set of 100 messages was manually tagged and serves as a Gold Standard for testing the tagging of events and temporal information.

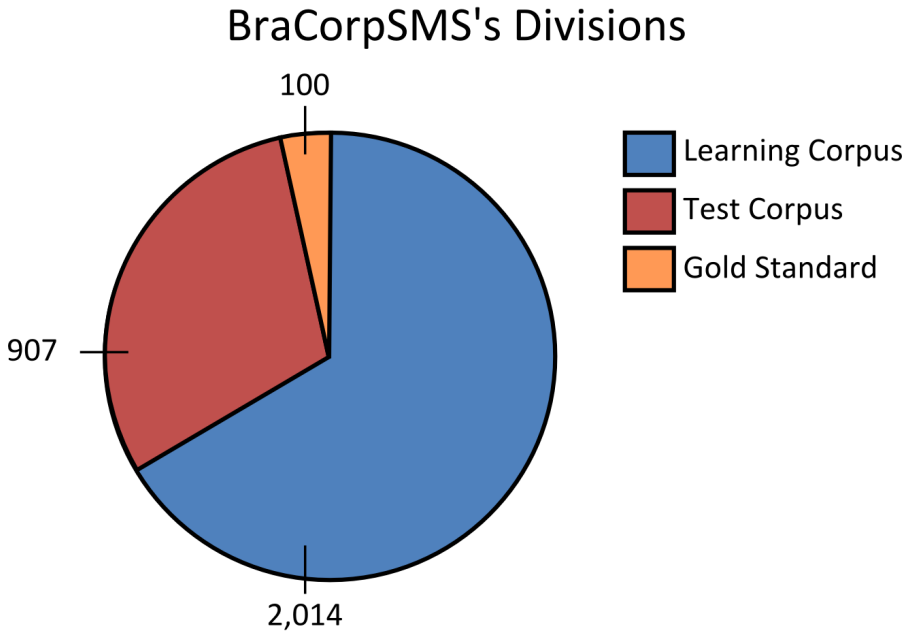


Figure 5.4 – BraCorpSMS's Divisions

As seen in Figure 5.4, from the 3,021 initial messages in BraCorpSMS, the Learning Corpus comprises 2,014 SMSs. Accordingly, we reserved 100 messages for the Gold Standard and the Test Corpus ended up with 907 messages.

5.2 Prototyping a System Based on the Architecture

In this section, we describe the prototyping of the IE system that aims to validate the architecture proposed in the previous chapter. We present relevant aspects of each component and real examples of input and output when applicable.

5.2.1 Prototype Overview

We prototyped the architecture using Python[1] as programming language mainly due to its ease of use, productivity and features for handling strings, lists, tuples and dictio-

[1] https://www.python.org/

naries. We decided to use version 2.7 as it is compatible with the majority of the applications and Python libraries available.

For the same reason, we chose the Natural Language Toolkit (NLTK[2]), designed for Python, in its version 3.0. Despite being tailored for standard texts, NLTK provides some interesting features for Portuguese that we thought would be useful for our process, like tokenizers, stemmers, Part-of-Speech Taggers, and annotated corpora (Floresta Sintá(c)tica and MacMorpho) for training purposes.

5.2.2 Linguistic Processing Steps

The Linguistic Processing component is the first step of the IE system. The input of this component is the original message "as is" in the corpus and its output is a linguistically processed message, properly split in sentences, tokenized and POS tagged. The component is responsible for handling each message following six steps (as shown in details in Section 4.3): Normalization, Sentence Splitting, Tokenization, Stopword Removal, POS Tagging, and Spell-checking. Next, we highlight the actions that make part of this prototyping.

Normalization

The normalization step comprises standardizing text input. The messages contain many misspellings, as texters tend not to follow spelling and grammar rules, which led us to address this matter beforehand, aiming to cover the most common cases found on the Learning Corpus. For instance, there can be many different ways to write a same word. To illustrate this problem, Table 5.3 shows sixteen spelling variations for the word ‘desde’ (since) found in short messages in the Learning Corpus. It is noticeable that some of these variations are caused by different levels of literacy, besides idiosyncratic SMS language characteristics.

Table 5.3 – Spelling variations for the word ‘desde’ (since)

Spelling variations			
desd	desda	deus de	deis de
ds da	dese	dez de	des da
des do	des das	deus da	ded
dsd	derde	desde de	dese as

As seen in Table 5.3, dealing with every possible word variation is challenging, if not impossible. We created a set of regular expressions in order to deal with a restricted range

[2]http://www.nltk.org

of variations of the most frequent terms found. Although not optimal, this approach can be incremental and allows us to keep the intended meaning of the messages. For example, we normalized all instances of "na", "nas", "no" and "nos" to "em".

In this step, we lowercased the messages, removed commas, hyphens and special characters, such as '#' and '@'. In order to discard unnecessary full stops, such as in zip codes or abbreviations, we used regular expressions, removing or replacing them by whitespaces.

Sentence Splitting, Tokenization and Stopword Removal

After removing unnecessary punctuation marks, we split sentences considering periods, exclamation marks or question marks as valid end marks. We do not consider commas to split sentences because this may split related words, important for temporal expression recognition or event detection. Moreover, we wanted to preserve sentences, in order to analyze the intended meaning in its entirety. We consider that if a temporal expression and an event occur in the same sentence, it is likely that they are related.

Next, each individual sentence passes through a tokenization step, using whitespaces to mark word boundaries. The prototype uses a NLTK built-in regular-expression-based tokenizer, wordpunct_tokenize[3], which splits the string into a list of tokens.

Aiming to avoid processing of words of little value for information extraction, the prototype uses a list of stopwords. Even though NLTK comprises a list of stopwords in Portuguese, we decided not to use it for it contains words that can be an important part of temporal expressions or events, such as "mais", "há", or "estamos".

We built a list comprising 45 stopwords we gathered while studying the Learning Corpus. The list consists of an external text file where words are separated by a semicolon. Besides the common stopwords in Portuguese, like 'para', 'o', and 'a', we considered common word shortenings and phonetic abbreviations, such as 'vc' and 'q'. The main advantage of keeping the list of stopwords as an external file is that, if necessary, it can always be enriched.

Part-of-Speech Tagging

This step resorts to the NLTK built-in taggers, which in turn require a tagged training corpus. The toolkit comprises two corpora - MacMorpho[4], a Brazilian Portuguese POS-tagged news corpus, with over a million words from journalistic texts extracted from the newspaper Folha de São Paulo, and the Floresta Sintá(c)tica corpus, which contains texts in Portuguese (from Brazil and Portugal) automatically annotated by the Palavras parser [13].

[3] http://www.nltk.org/api/nltk.tokenize.html
[4] http://www.nilc.icmc.usp.br/macmorpho/

We performed a few tests in order to select the most adequate to train the tagger. We selected 100 SMSs at random from our Learning Corpus and preprocessed them according to steps already presented. We compared the output using MacMorpho and Floresta as training corpora (detailed results can be seen in Appendix B). For that comparison, a negative score was assigned every time a wrong tag was attributed to a token. Likewise, a positive score indicate a correct tag attribution to a token.

Both taggers trained with the corpora encountered difficulties when dealing with the SMS language. This is understandable, as both training corpora comprise texts in standard Portuguese. Final accuracy scores were quite similar (68.68% for MacMorpho and 68.45% for Floresta), but MacMorpho presented limitations when dealing with proper nouns, which led us to opt for using Floresta. The tagger converts each token to a tuple containing the token and its corresponding POS tag (as seen in 2.1.1).

Spell-checking

The lack of a tagged corpus of texts in SMS language hampers the POS tagging step. Even though the Normalization handles some level of misspellings, according to a set of predefined rules, many words not written in standard Portuguese remain untagged, which can pose a problem towards event detection, since the Event Processing component relies on POS tags (this component is explained in more details in Section 5.2.3). To address that matter, the IE Architecture comprises a Spell-checking step, in order to mitigate the spelling variation problem.

To address that matter, we tested PyEnchant[5], a spell-checking library for Python, which comprises dictionaries for British and American English, German and French by default, but allows users to install external dictionaries. Therefore, we added the Open Office[6] Brazilian Portuguese dictionary extension. PyEnchant provides an easy-to-use interface that allows checking the spelling of words and getting a list of suggestions for misspellings. To avoid affecting the prototype's performance, the prototype only checks the spelling of untagged tokens.

This step also checks whether the most likely suggestion receives a proper POS tag. If so, it replaces the untagged word for the suggested one. If not, the original word remains unchanged. For instance, the misspelling "post" would be changed for "poste" (pole), as, according to the library, this suggestion is the most likely to be correct. The prototype assigns the corresponding POS tag to a noun ("n") and makes the exchange.

There are some special cases of domain-related words the POS tagger cannot resolve, like "transformador" (transformer), "estouro"(burst), "enchente"(flood), etc. It is likely that these cases were not in the training corpus of the POS tagger. We added them to an

[5]https://pythonhosted.org/pyenchant/
[6]http://www.openoffice.org/

external list, along with their corresponding POS tag, so the prototype can use this list to review untagged tokens.

Furthermore, this step also comprises revising tokens tagged as 'num' and converts numbers in text format to numerals. For example, "vinte e quatro" (twenty-four) is replaced by '24', "dez" (ten) is replaced by '10' and so on. For that issue, we listed word variations for units, tens and hundreds.

Figure 5.5 shows an example of how the Linguistic Processing component operates. The upper box represents an unprocessed message, in the XML format of BraCorpSMS. After passing through the seven processing steps, the component outputs a list of tuples containing tokenized sentences and their POS tags along with an identifier of the original message and its delivery date. It is worth mentioning that, in this example, the prototype divided the sentence in two, as there is a period between "SEGUNDA" and "#IDNUM".

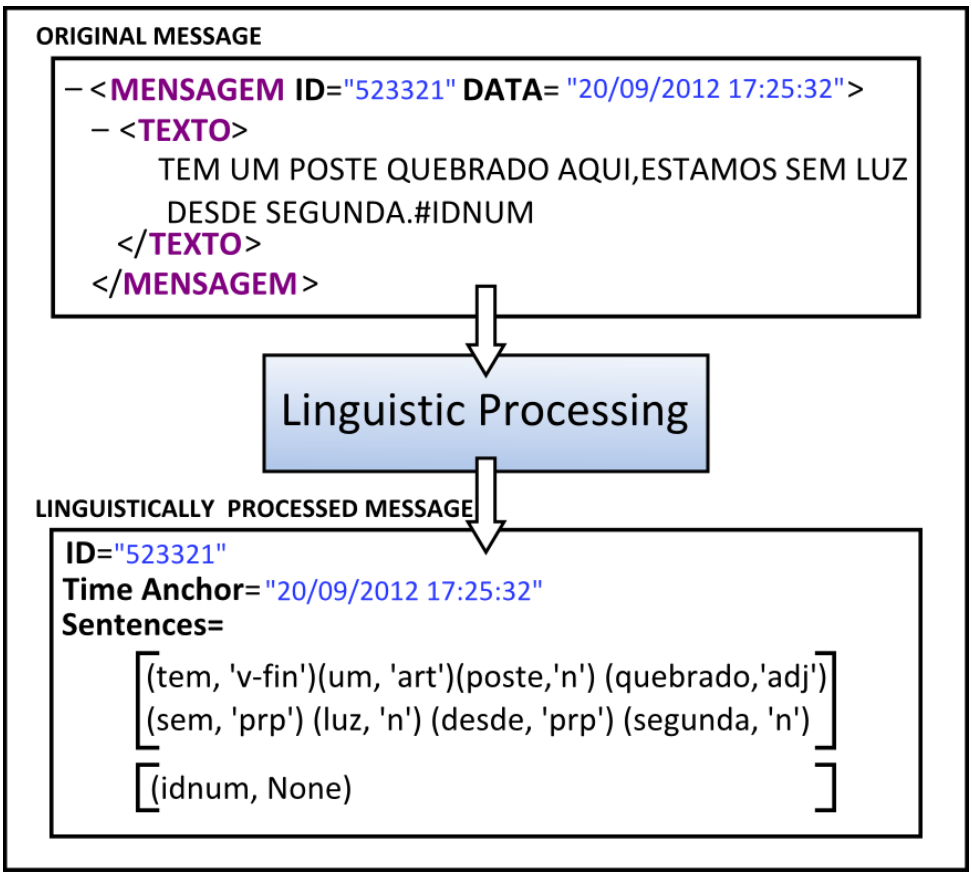


Figure 5.5 – Example of Linguistic Processing Input and Output

5.2.3 Temporal Processing

Once tagged, the messages proceed to the Temporal Processing component. As illustrated in Figure 4.4, the component comprises the following steps: a Temporal Expres-

sion Recognizer, a Temporal Reference Classifier and a Temporal Tagger. There are two external resources associated with the component: a set of regular expressions and a list of lexical triggers. Here, we highlight the main aspects of this component's implementation.

Temporal Expression Recognizer

Starting from a linguistically processed message containing a time anchor and the processed text sentences, the Temporal Expression Recognizer must be able to identify and extract existing temporal information. For instance, the duration of an event, such as a power outage, may be of great importance to indicate the severity of the problem. A client may inform the existence of a natural disaster that causes a blackout that lasts for hours and affects an entire region.

However, since texters express themselves in many different ways, we may find many variations for the same temporal information like '10:30', '10h30', '10h30min', and so on. In light of this, we resort to a rule-based matcher, which relies on a set of regular expressions. These rules cover the variations found in the Learning Corpus. Table 5.4 shows some examples of rules created. Moreover, we added rules to identify days of the week (segunda, terça, quarta, etc.) and times of the day (manhã, tarde, noite and madrugada).

Table 5.4 – Examples of regular expressions comprised by the Temporal Expression Recognizer

Regular Expression	**Examples**
([0-9]{1,2})(\s)*(h|hra*)(s)*	1h, 2 hr, 9hrs, 10hra, 11 hras
([0-9]{1,2})(h|:)([0-9]{1,2})	10h00, 9h30, 11h15, 10:00, 9:30
([0-9]{1,2})(\s)*(min(uto)*(s)*)	10min, 20 min, 10mins, 20 mins, 10minutos, 20 minutos
([0-9]{1,2})(\s)*(dias*)	2dia, 10 dia, 2dias, 10 dias

Since incoming messages express situations that happened or are happening, the existing temporal expressions refer to past or present events. In order to identify more complex expressions like 'desde ontem às 14h', we built a list of lexical triggers, containing the most common temporal keywords found in the Learning Corpus. Examples of these keywords can be seen in Figure 5.6.

The recognizer starts from a matched rule in search of the extent of the temporal expression. It makes a greedy search comparing tokens surrounding the expression to the list of lexical triggers. Tokens that correspond to words in the list are considered part of the temporal expression. Once the boundaries of the temporal expression are identified, the temporal expression may be classified.

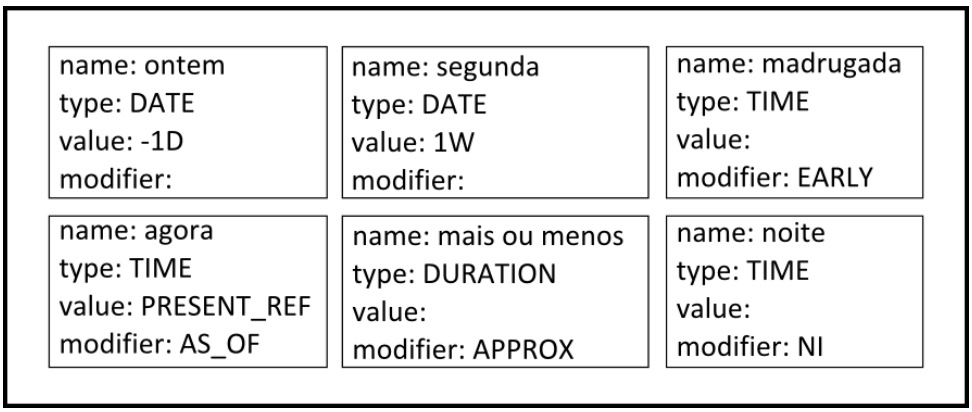


Figure 5.6 – Examples of Lexical Triggers

Temporal Expression Classifier and Tagger

Given the emergency nature of the communication, and according to the categories defined by Kevers (as presented in Section 2.1.2) we treat temporal expressions that indicate precise or fuzzy durations of events that are happening or already happened. So as to define a tagging standard, we use the TimeML specification language, given that it is the standard for most temporal expression recognition systems [20, 40]. Even though the current TIMEX version is the TIMEX3, we opted on using TIMEX2 instead of TIMEX3, as the guidelines to the latter indicate that the extent of temporal expressions should be as small as possible [54]. For instance, according to TIMEX3, "Sexta-feira às 20h" (Friday at 8:00 p.m.) is recognized as two time tags, "Sexta-feira" and "20h". In order to process temporal expressions associating them to detected events, we want to preserve expressions in a single tag. Appendix A presents more details on TIMEX2 tags.

Following TIMEX2, we can tag temporal expressions indicating dates, times or durations. Each tagged temporal expression is assigned a value, a modifier or a value and a modifier, according to its type. With the value and time unit of a temporal expression, along with a time anchor (in our case, the delivery date), it is possible to calculate the beginning or duration of an event. For instance, the lexical trigger "ontem" (yesterday) has a type 'DATE' and value -1D, indicating the amount of one day must be diminished from the time anchor in order to determine the TIMEX value. Likewise, other expressions such as "minutos" (minutes) or "dias" (days) indicate the amount of time that must be decreased from the time anchor. Accordingly, the entry "noite" (night), which has a modifier 'NI', adds it to the value of the TIMEX. We extended the list of lexical triggers to comprise the type, value, and modifier of each temporal keyword. Figure 5.6 shows examples of lexical triggers, their types, values and modifiers. The Temporal Expression Classifier consults this list to determine the modifiers and values of each expression. Since lexical triggers inside the same time expression can be of different types, the Tagger must deal with ambiguities. For that matter, it considers first lexical triggers expressing largest periods of time. Duration expressions have precedence over dates and dates have precedence over times. Next, the Tagger

is responsible for grouping all values and modifiers in a single form and then assigning the corresponding time tag to each analyzed temporal expression.

Figure 5.7 shows an example of how the Temporal Processing Component operates. Considering the same example and the output used for the Linguistic Processing Component, the Temporal Processing Component analyzes both sentences in the preprocessed message. As the second sentence does not contain a temporal expression, we focus on the first one. The rule-based matcher begins identifying "segunda" (Monday) as the core of the temporal expression. Then, the Temporal Expression Recognizer determines "desde segunda" (since Monday) as the extent of the TIMEX, since "desde" is a lexical trigger. The Temporal Expression Classifier indicates that "desde" represents a duration, of modifier 'START'. On the other hand, "segunda" represents a date of value 1. Then, it compares the day of the week from the time anchor (September 20 - a Thursday) to "segunda", which is the starting point, resulting in a period of 4 days. Since the type duration has precedence over dates, the Classifier defines the type as duration and the value as 'P4D', indicating the 4 day period. Finally, the Tagger receives the type and value of the expression, groups and adds them to a TIMEX tag on the corresponding temporal expression.

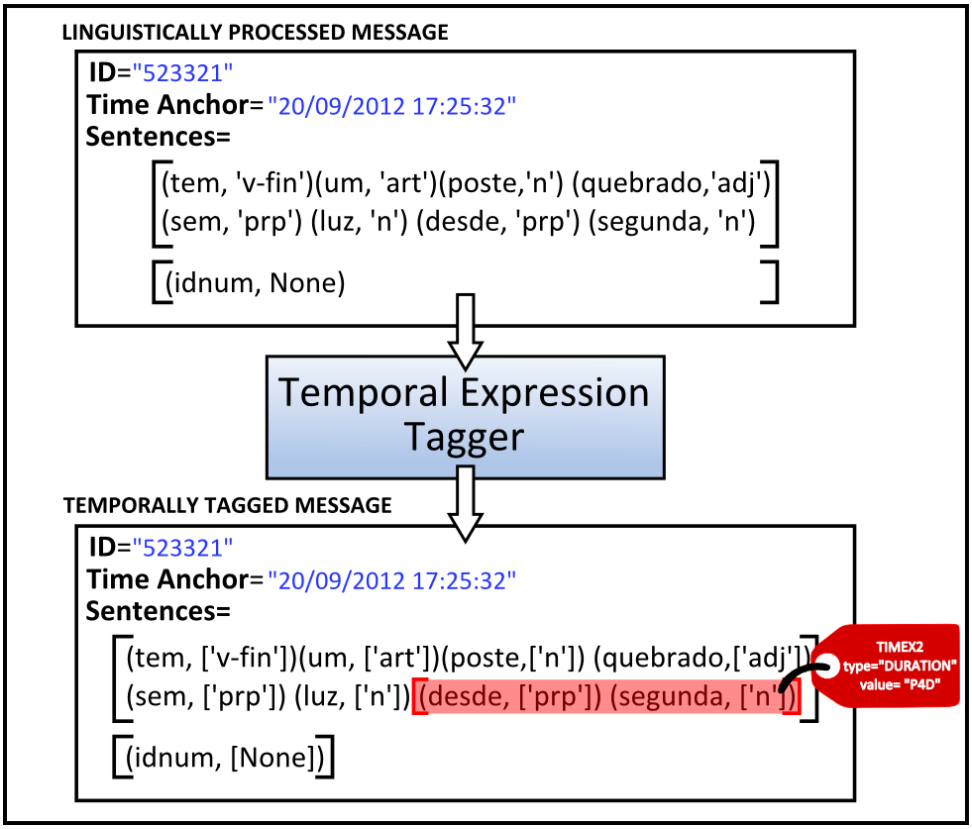


Figure 5.7 – Example of Temporal Tagger Input and Output

5.2.4 Event Processing Component

The next component of the IE Architecture is the Event Processing, which as illustrated in Figure 4.5 consists of an Event Detection step, an Event Classifier and a Tagger. Besides, the component comprises a structure validator to support event detection and a list of domain-related categories, which helps determining events of interest and keywords related to them. We detail each step as follows.

Event Detection

To detect events, we focused on a verb-triggered rule-based approach. Mario Perini, in 'Gramática do Português Brasileiro' states that "the verb is the key to the Portuguese language syntax" [38]. From understanding the verb, its meaning and accessories, one can determine the structure of the sentence of which it is part. According to the same author, although there is no list of all existing sentence constructions in Portuguese, a native speaker has a list of possible constructions, most of which consider the structure: Noun Phrase + Verb + Object. Both the Noun phrase and the Object can play semantic roles of agent and patient. Also bearing in mind that, in Portuguese sentences, the subject can be omitted, we considered the constructions in Table 5.5 for our event detection step.

Table 5.5 – Sentence Constructions (adapted from [38])

Sentence Constructions
Noun Phrase + Verb + Object
Noun Phrase + Verb
Verb + Object (Null Subject)
Adverb + Verb + Object (Negative Sentences)

It is worth mentioning that, in Table 5.5, we do not distinguish agents from patients.

From this model, the prototype iterates through sentences looking for the POS tags assigned during the Linguistic Processing in order to find verbs. The tagset used by the prototype can be seen in Section 2.1.1.

Similarly to the Temporal Expression Recognizer, after finding a verb, the Event Detection step marks the boundaries of the event mention. For that, it checks the surroundings of the verb, as long as the tokens are in the list of event-related words, looking for nouns, prepositions, noun compounds, adjectives and pronouns. While at this, the component also checks for compound verbs, but considers verbs interspersed with other POS tags as different event mentions. Moreover, the Event Detection step considers adjectives and nouns (or noun compounds) the boundary of the event mention.

Next, it verifies detected events, in order to assure they are valid structures. To address that matter, we built a "Structure Validator", a state diagram representing valid

sequences of POS tags. This diagram reflects the Learning Corpus and it was reviewed according to the Portuguese Grammar [38]. Figure 5.8 shows the state diagram.

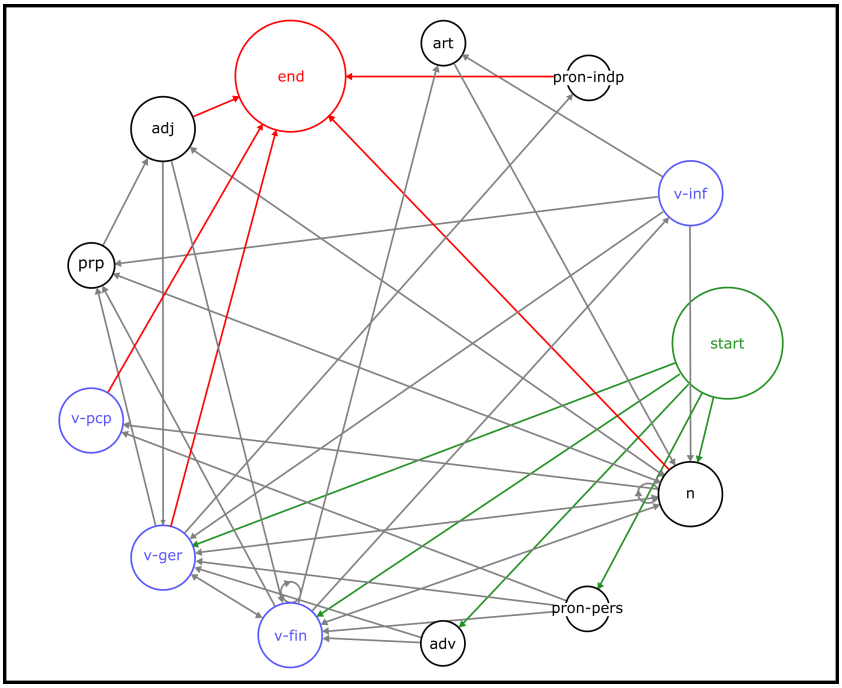


Figure 5.8 – State Diagram representing valid structures in the application

The Structure Validator searches for a path through the diagram that comprises the same tag sequence as the detected event. If this path is found, then it is considered a valid structure and the component proceeds to classify the event. If not, the event candidate is discarded. For instance, the sentence "Estou sem luz" (I have no electricity) is tagged with the sequence 'v-fin', 'prp' and 'n'. As we can see in Figure 5.8, this tag sequence constitutes a valid path in the state diagram: therefore it is considered a valid structure.

Event Classifier and Tagger

As mentioned in Section 2.1.3, the event detection and recognition task is context-dependent. In other words, to identify relevant events, domain knowledge is necessary. In an extensive study of the Learning corpus, we observed the clients communication during emergencies as well as what they are notifying. From this, we listed the most relevant events we found whilst defining the annotation standard. We selected the most frequent words in the corpus to help identify domain-related information. This observation led us to propose three non-mutually exclusive categories of events, regarding the scope of the event and thirteen notification types that provide additional situation awareness information, allowing us to establish a scale of severity between events. Table 5.6 details the categories of events and the notification types considered.

As exposed in Table 5.6, each category covers different scopes. "Instalação" refers to messages containing information regarding the consumer unit (electrical installation).

Table 5.6 – Categories of Events

Category	Notification Type	Severity Degree
Instalação	Falta de Luz	3
	Oscilação	2
	Meia Fase	2
Rede	Queda de Fios e Cabos	4
	Queda de Transformador	4
	Chave Fusível Desligada	2
	Curto Circuito	4
	Queda de Poste	5
	Fogo na Rede	5
Ambiente	Queda de Árvore	4
	Raio	3
	Vento	3
	Chuva	3

"Rede" groups information about the electrical grid status and its components. "Ambiente" comprises information regarding the environment that might affect the electrical grid, such as fallen trees, storms and lightning.

In order to properly classify the event, we split the sentence and analyze separately the noun phrases, verbs and objects, in search for domain-related words. For this task, the component counts on a list built from 83 words related to the notification types, collected from the following sources:

- Dicionário Criativo[7], a dictionary that searches words in different databases, such as Aulete Digital[8] and Wikipedia;
- WordNet[9], a large lexical database of English and its version in Portuguese, WordNet.Br[10];
- Tep 2.0[11], a Brazilian Portuguese thesaurus;
- Dicionário de Sinônimos Online[12], an online dictionary of synonyms for Brazilian Portuguese containing over 30,000 synonyms.

The classifier checks each part of the sentence (noun phrase, verb and object), comparing tokens to the list of domain-related words. The classifier looks for agreement. For instance, the sentence "caiu uma árvore na rede" (a tree fell over the power grid), is

[7] http://dicionariocriativo.com.br/
[8] http://www.aulete.com.br/
[9] http://wordnetweb.princeton.edu/
[10] http://www.nilc.icmc.usp.br/wordnetbr/index.html
[11] http://www.nilc.icmc.usp.br/tep2/
[12] http://www.sinonimos.com.br/

divided in two phrases after the processing: "caiu" (verb), and "uma árvore em rede" (object). Once verified, the Event Classifier considers that while the verb cannot determine a notification type, the object by itself indicates the notification type "Queda de Árvore", due to the presence of the words "árvore" and "rede".

The classifier determines the severity degree of the event according to its notification type. The severity degree for each notification type is exposed in Table 5.6. Whenever there are more notification types in the same event mention, the classifier assigns the event with the highest degree among the notifications found. For example, an event mention notifying "Chuva" (severity degree 3) and "Curto Circuito" (severity degree 4) would be classified as degree 4.The severity degrees were established with the aid of domain-knowledge experts. More details on this activity can be seen in Section 5.3.1.

Finally, the Tagger groups all the information in a set of tags, according to its categories, notification types and severity degrees. For example, the sentence ‘caiu arvore fio caido’ notifies a fallen tree and a downed power line at the same time. Therefore, the sentence receives the tags <REDE cond=’queda de fios e cabos’> and <AMBIENTE cond=’queda de arvore’>, where ‘cond’ stands for condition. Both notification types have the same severity degree (4), therefore the event is assigned severity 4.

Figure 5.9 illustrates how the Event Processing Component operates. In the first sentence, the rule-based matcher identifies "tem" (there is) as the core of the event mention, as it contains a POS tag indicating a verb (v-fin). Next, the event detection step defines the boundaries of the event, stopping at the adjective. The result, "tem um poste quebrado" (there is a broken power pole), goes through the Structure Validator, which indicates that this tag sequence (’v-fin’, ’art’, ’n’, ’adj’) is valid. The Event Classifier separates the event in two, verb (’tem’) and object (’um poste quebrado’), and compares the tokens to the list of domain-related words. The tokens ’poste’ and ’quebrado’ lead the classifier to define the event as a notification of a broken power pole, with a severity degree of 5. As there is no other notification type involved, the Tagger attaches the tag <REDE cond=’queda de poste’ severity=5> to the detected event.

5.2.5 Information Fusion

Once the information are already tagged, one can use different approaches to visualize and understand such data. Here we comment on how we proceed to fuse the extracted information being aware of other possibilities that could be explored in a more extended study. It is worth mentioning that some of the previously presented visualization approaches, such as Ritter *et al.*’s calendar [41], are not applicable to our case study. In their work similar event mentions refer to the same event. In our case, event mentions in different messages may refer to different events.

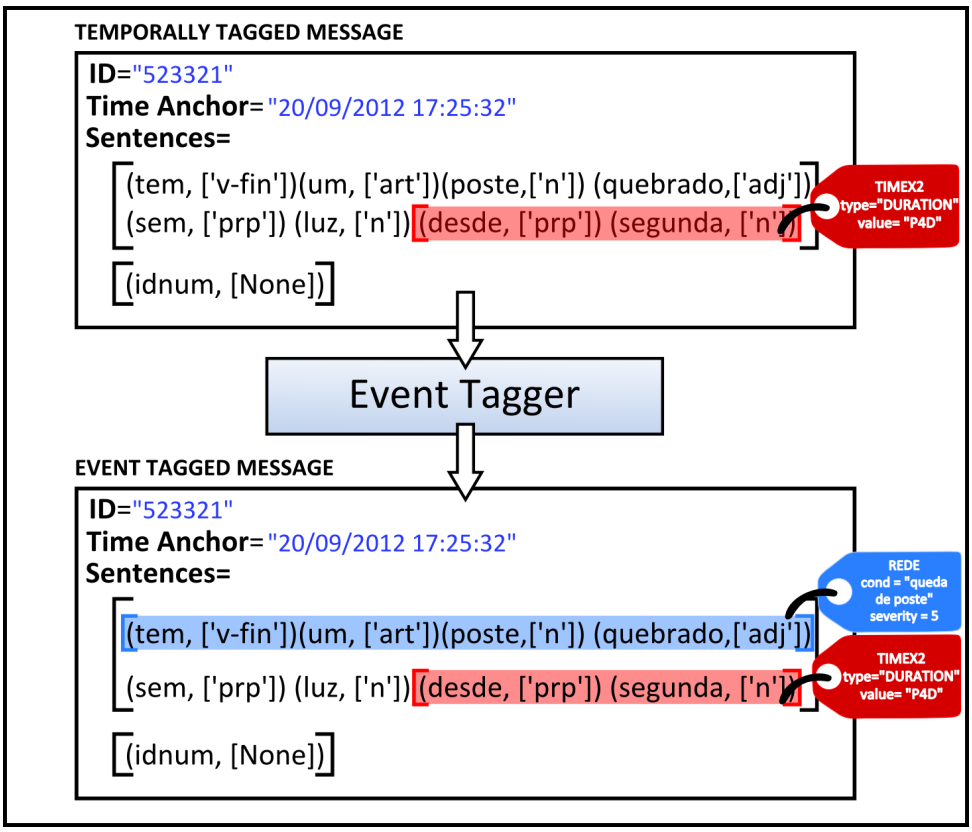


Figure 5.9 – Example of Event Processing Component Input and Output

To address this issue, we exported the output of the prototype to a spreadsheet containing the messages and their corresponding notification types and severity degrees, allowing us to generate a series of charts. From these charts, one can visualize the application of the proposed model. Structured information can be more easily manipulated, in order to speed up the recognition and attendance of occurrences with more situation awareness.

Despite the existence of other possible approaches, spreadsheets are a quick and simple alternative to explore data. Moreover, this component may be improved along with the expansion of the model of categories of events. Through these charts, we have more tools to analyze the prototype behavior, *i.e.*, its mistakes, successes and areas for improvement. We will explore more on this in the following section.

5.3 Evaluating the IE Results

In this section we detail the steps taken towards evaluating the prototype. We present the evaluation plan, its execution, and discuss the results.

5.3.1 Evaluation Plan

Upon the need of evaluating the prototype's IE components, it is essential to conduct experiments and assess their results. For that reason, we elaborated a three-step evaluation plan comprising: confirming the model of categories and notification types; providing a Gold Standard - a manually annotated corpus considered as "definitive answer"; and comparing the prototype's results to the Gold Standard. Furthermore, we assess the results of the Information Fusion component over the Gold Standard and the Test Corpus.

Some of these answers shall be provided by domain experts, which led us to invite three judges with different know-how over the same domain. Two of the judges are electrical engineers, while the third one is a software engineer responsible for maintaining the current SMS Message processing system.

To obtain their contribution, we elaborated two questionnaires using the Google Forms[13] platform, which allowed us to create and share forms via web. The first questionnaire aimed at gathering the judges' opinion about the model of categories of events. After establishing the model, the judges received a second questionnaire, comprising 100 SMS messages in which they should indicate the existence of any of the predefined notification types as well as temporal expressions. These answers were used to compose the Gold Standard. As an evaluation task, we compared them to the output of the IE prototype to obtain Precision, Recall and F-measure. Through sections 5.3.2 to 5.3.5 we detail each step taken and the evaluation results.

5.3.2 Confirming Categories of Events and Notification Types

The first questionnaire aimed to validate the categories of events and notification types predefined in Section 5.2.3. The assistance of the domain experts was necessary to validate the model and to determine the degree of severity of each notification type, which is important to rank relevant information and classify detected events. The questionnaire can be seen in Appendix C.

Each question corresponded to a specific notification type. Furthermore, judges should assign a value between 1 and 5 indicating the severity degree of each notification. Blank spaces were left available at the end of the form so judges could contribute regarding modifying the model. The judges validated the model and suggested the addition of an "electricity theft complaint" type. The results are shown in Table 5.7, in columns 2, 3, and 4.

In order to measure the inter-rater agreement, we used the kappa statistic. Kappa measures the agreement degree beyond the expected by chance. Kappa values can vary

[13] http://www.google.com/forms/about/

Table 5.7 – Judgment assigned to the notification types

Notification Type	Judge 1	Judge 2	Judge 3	Average
Falta de Luz	5	4	1	3
Oscilação	3	2	1	2
Meia Fase	2	2	1	2
Queda de Fios e Cabos	5	4	4	4
Queda de Transformador	4	4	3	4
Chave Fusível Desligada	1	4	3	2
Curto Circuito	5	3	4	4
Queda de Poste	5	5	4	5
Fogo na Rede	5	4	5	5
Queda de Árvore	4	4	4	4
Raio	2	3	4	3
Vento	2	4	4	3
Chuva	2	3	3	3

from 1 (representing a complete agreement) to 0 or below (indicating no agreement or agreement by chance) [49]. Equation 5.1 shows how to calculate the kappa coefficient [5]. P(A) is the relative observed inter-rater agreement, while P(E) is the probability of agreement by chance.

$$K = \frac{P(A) - P(E)}{1 - P(E)} \tag{5.1}$$

We used the results to assess the range of scores given by the judges. Considering a 95% confidence interval, we obtained a value of K of 0.013, resulting in a poor level agreement, which might indicate that it would be more appropriate to use fewer degrees of severity.

By analyzing the results individually, we noticed that, in most cases, there was little variation between the scores given. To avoid another judgment round, we decided to use the average of the assigned scores, resulting in four severity levels, ranging from 2 to 5. This is due to the fact that there were only four scores '1', according to Table 5.7.

5.3.3 Gold Standard Creation

As mentioned earlier, the Gold Standard is required to evaluate the prototype's performance. Thus, we have prepared a questionnaire with the part of BraCorpSMS reserved for the Gold Standard (as seen in Section 5.1.5). This questionnaire aimed at providing a manual tagging for the future evaluation of the prototype's temporal and event taggers. Each question comprised a different SMS message (anonymized), containing 14 non-exclusive options (thirteen for notification types and one for temporal expressions). An 'Others' option was also available, so the judges could express disagreement with the existing alternatives whilst offering their judgment. The first page of the questionnaire, containing the orientation and a sample question can be seen in Appendix D.

Since Google Forms provides the results in an Excel spreadsheet, we had to deal with this its features to create the Gold Standard. By default, all answers were arranged in rows, where each cell corresponds to an answer and the text of every option in a comma-separated format. It means that, if a judge chose the options 'Informação Temporal' and 'Instalação - Falta de Luz', then the corresponding cell would contain the string 'Informação Temporal, Instalação - Falta de Luz'.

Accordingly, we then adapted the spreadsheet to compare the strings and to check if at least two (so the majority) of the judges agreed on tagging the same events or regarding the mention of a temporal information. In the manual analysis by the judges, only one message was object of disagreement, and therefore left untagged, but kept in the Gold Standard and considered as a message without relevant information. After, we converted each option to an integer number (1 for 'Informação Temporal', 2 for 'Instalação - Falta de Luz', and so on), in order to facilitate the comparison between the Gold Standard and the prototype's results. We used 0 to represent an untagged message. Finally, we grouped the messages identifier with the list of numbers indicating the correct answers, separated by commas, and exported them to a text file.

5.3.4 Evaluating the Prototype's Taggers

To evaluate the prototype's taggers performance, we used its output to generate a file with the same format of the Gold Standard. Both outputs are detailed in Appendix E. After, we developed a simple Python script to compare both files. The performance evaluation was given by Recall, Precision and F-measure, values exposed in Section 2.3.

From that, we obtained Precision of 88%, Recall of 59% and F-measure (F1) of 71%. After analyzing the results in details, we observed that the prototype performed well on correctly identifying relevant events, with 125 true positives over 16 false positives and 84 false negatives. This indicates that the set of defined rules is accurate while detecting the events and temporal information mentioned in the Gold Standard corpus. However, a low Recall score alerts us that there are other events, domain-related words and temporal information still uncovered by the model created. Furthermore, it also indicates that part of the information in the corpus remains unidentified.

As improvement opportunities unveiled, we could mention resorting to a more appropriate tagged corpus trained over the SMS language. This resource would decrease the number of untagged tokens, which in turn would increase the accuracy of the event detection step. However, to the present, we do not know of the existence of such resource for the Portuguese language. Moreover, we must learn on where the prototype failed and update our model as well as the prototype's external resources over the uncovered information.

Table 5.8 shows the hit percentage of the prototype when compared to the Gold Standard, as well as the amount of all notifications found in this corpus. The prototype could not resolve mentions to "Meia Fase", "Oscilação", "Queda de Transformador", "Raio", "Vento" and "Chuva" events. Analyzing the messages from the Gold Standard, we can see that some mentions of events were not detected by the prototype as they omit verbs, as in "toda nossa comunidade sem luz devido muita chuva ventos fortes" (our entire community without electricity because of a lot of rain and strong winds). There were also problems in differentiating "está" (is) and its misspelling "tá" (absent in the training corpus) from "esta" (this) which compromised the detection of some events.

In addition, the Temporal Processing component obtained good results over the Gold Standard. In fact, in one of the evaluated messages, the component tagged "15 minutos" (15 minutes) as a temporal expression, while the judges did not recognize it. On another case, the judges tagged the message "luz estamos sem lus hs 110" (we do not have electricity hs 110) as a temporal expression, although this understanding is not clear even for humans.

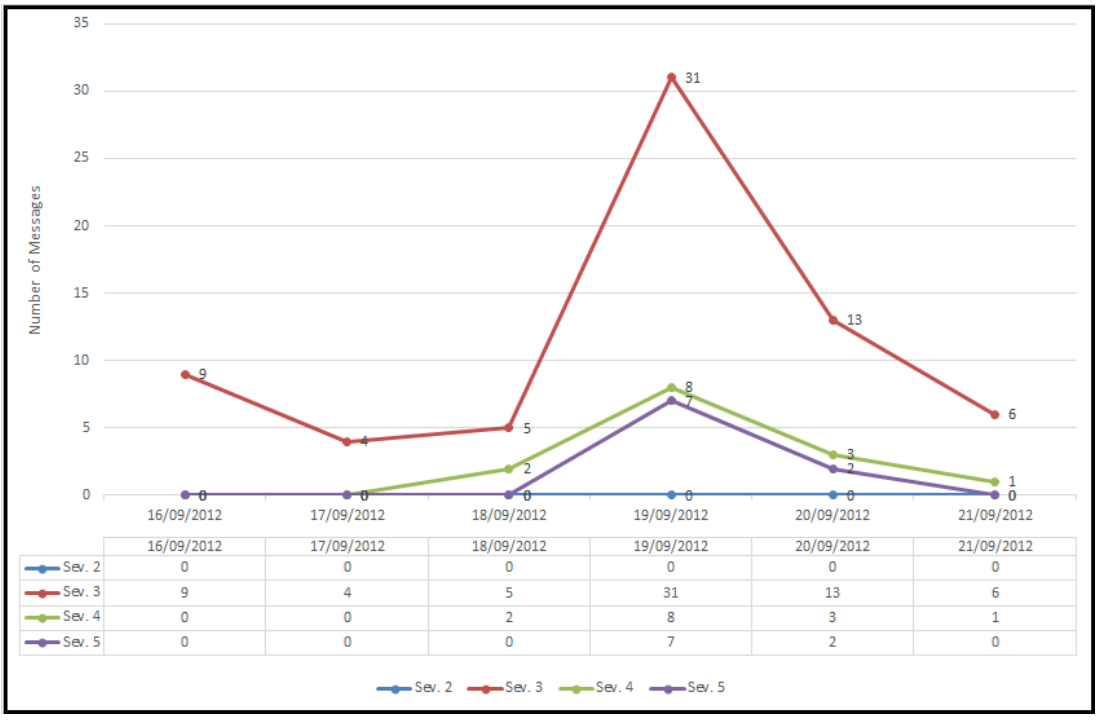
Table 5.8 – Hit Percentage by information type

Extracted Information	**Prototype**	**Gold Standard**	**Percentage**
Temporal Information	24	27	89%
Falta de Luz	86	88	98%
Meia Fase	0	9	0%
Oscilação	0	3	0%
Queda de Fios e Cabos	7	32	22%
Queda de Transformador	0	3	0%
Chave Fusível Desligada	1	5	20%
Curto Circuito	3	5	60%
Queda de Poste	4	10	40%
Fogo na Rede	5	7	71%
Queda de Árvore	5	12	42%
Raio	0	1	0%
Vento	0	4	0%
Chuva	0	2	0%

The low hit percentage of some results indicate that, whilst the verb-triggered event detection performed well over some notification types, such as "Falta de Luz", "Fogo na Rede" and "Curto Circuito", other approaches must be considered, in order to achieve a higher hit rate in event detection. Furthermore, Table 5.8 evidences the existence of additional information contained in the messages and still uncovered by the prototype.

5.3.5 Fusing the Information

With the intent of analyzing detected events based on some similarity level, we grouped messages by their severity degree and date of delivery. From that, we can visualize the relation between the number of messages received per day and their severity. Accordingly, we generated a chart from the extracted information. The Information Fusion component was tested with messages from the Gold Standard corpus, which is randomly generated from messages sent from 16 to 22 September 2012 and that compose BraCorpSMS. Figure 5.10 displays the component output. As the Gold Standard corpus ended up without messages from 22 September, we omitted this day from the chart. Notice that, during the period, there was no message with severity degree 2.



	16/09/2012	17/09/2012	18/09/2012	19/09/2012	20/09/2012	21/09/2012
Sev. 2	0	0	0	0	0	0
Sev. 3	9	4	5	31	13	6
Sev. 4	0	0	2	8	3	1
Sev. 5	0	0	0	7	2	0

Figure 5.10 – Number of Messages by Severity

Comparing the information, one can see that along with the increasing amount of messages (on 19 and 20 September) there is an increase in the degree of severity associated with the received messages. This perception is in line with the notion that more messages indicate more problems of serious nature. From this chart, a service provision team could observe the growth trend of severity and prepare for emergency situations.

In addition, we generated a second chart showing the notifications sent by customers according to the day of the week. Figure 5.11 shows this relation. As can be observed, not only the amount of power outage notifications increased, but others of higher severity degree increased on 19 September as well. We omitted the notification types with-

out corresponding messages, in this case, "Meia Fase","Oscilação", "Queda de Transformador", "Raio", "Vento" and "Chuva".

Next, we performed a second round of tests, over the Test Corpus, which comprises 907 messages. As shown in Figure 5.12, test results are similar to the previous, indicating an increase on severity degrees in 19, 20 and 21 September. Figure 5.13 shows the results separated by notification type, where we can see that over a larger corpus, the prototype was able to identify more mentions to different notification types, such as "Vento", "Chuva", "Oscilação", and "Meia Fase".

5.4 Chapter Summary

In this chapter, we presented a case study conducted in order to validate the proposed IE Architecture. We built BraCorpSMS from a set of 3,021 messages collected from an electric utility company according the process defined in Section 4.2. To study messages sent during emergency situations, we analyzed the data in a period of time where the amount of messages peaked.

After validating the process, we can confirm that it is feasible, but has some improvement opportunities, mainly because of the manual steps, which can be time-consuming. We were able to collect and filter, in an easy manner, text messages produced by costumers, so that these messages can be delivered to a temporal information and event extraction application. We did not face problems with transcription errors, as we collected the messages directly from the company's database. However, while applying other steps of the process, we faced some challenges. For instance, sensitive information was not standardized and, as we want to use the corpus for temporal expression extraction, we had to anonymize messages and guarantee the removal of all sensitive information whilst maintaining the intended meaning of the messages. Moreover, as the process does not include a normalization step, it was necessary to spend more efforts to address this issue while prototyping the IE architecture.

Afterward, we built a prototype in Python along with the NLTK toolkit. The prototype comprises a component for preprocessing a linguistic analysis, a component to tag temporal expressions, an event processing component and an Information Fusion component. Our first steps involved understanding the SMS language, which led us to divided BraCorpSMS in a Learning Corpus containing 2,014 messages, a Test Corpus with 907 messages and a Gold Standard composed of 100 messages set apart for evaluation purposes.

From this knowledge-based approach, we developed rules considering the SMS language to recognize temporal information mentioned in messages. As texters can express the same information in many different ways, we covered the spelling variations we observed on the Learning Corpus. Likewise, we looked for relevant event-related information in order

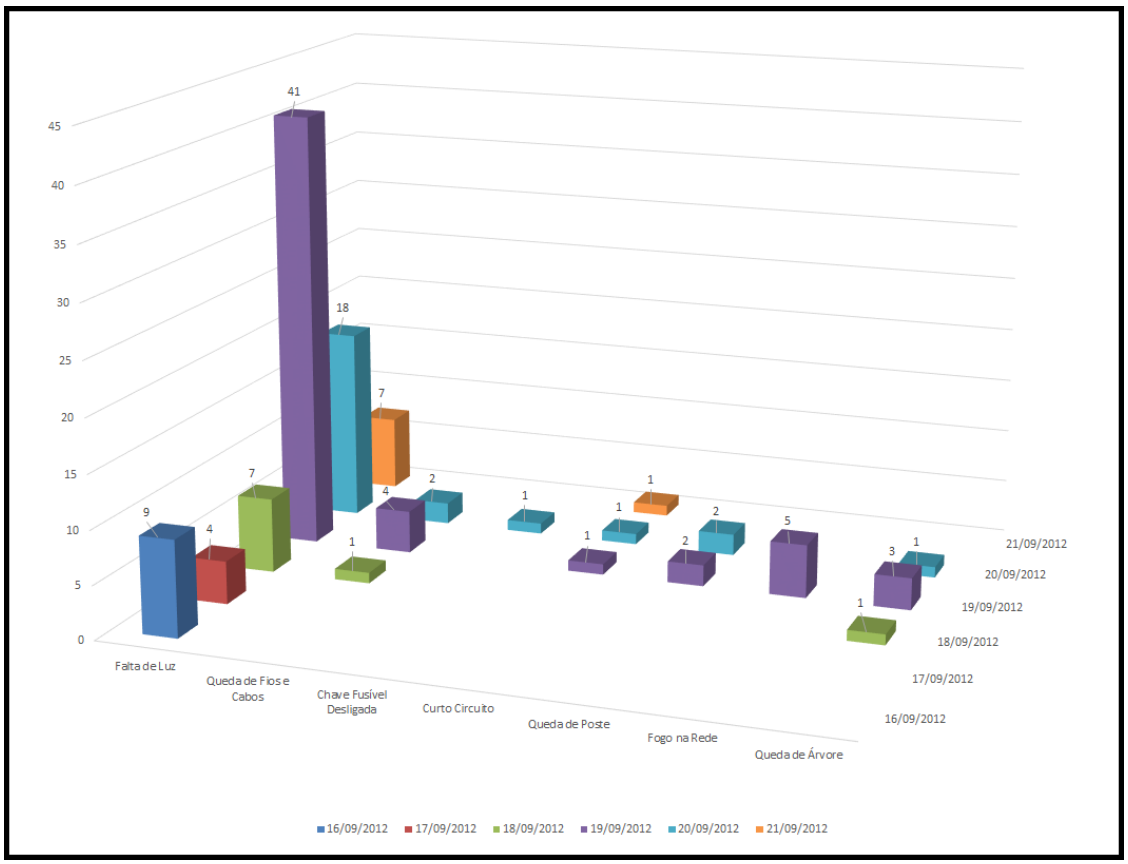
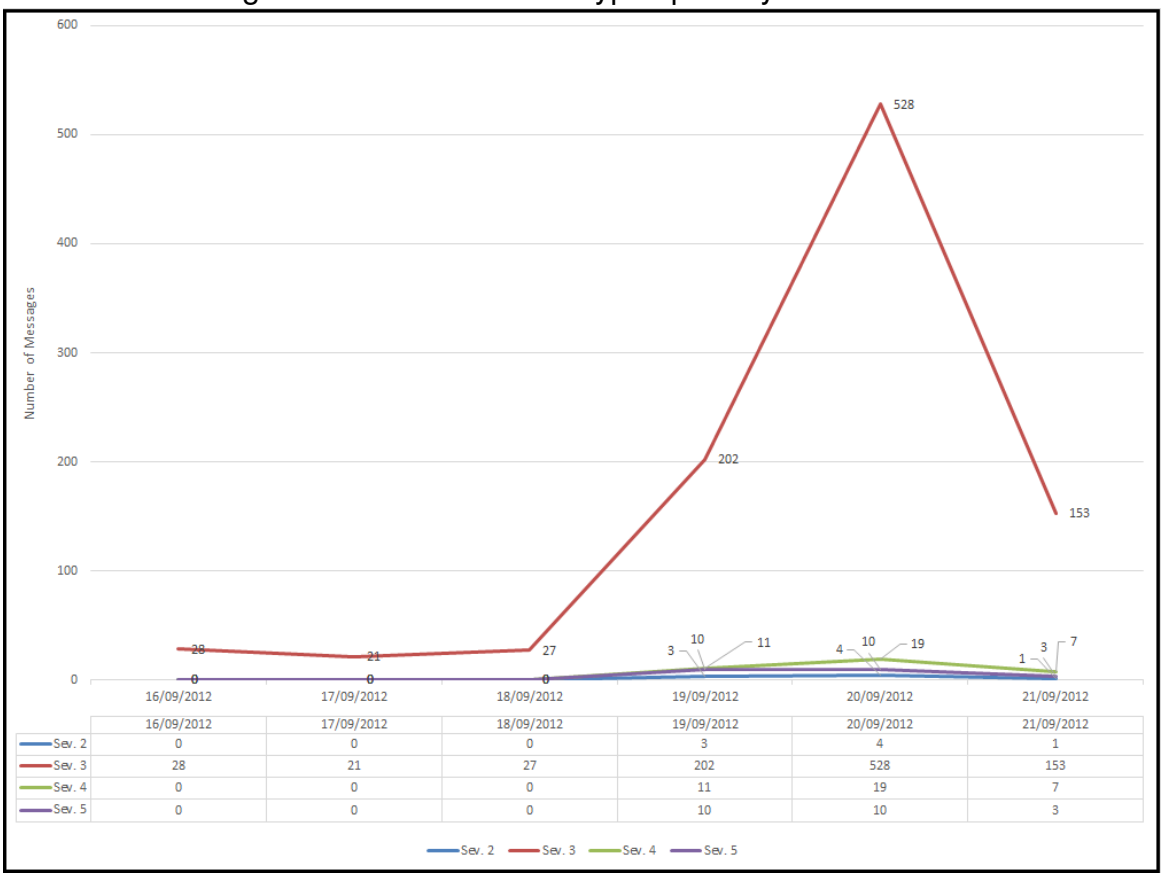


Figure 5.11 – Notification Types per day of the week



	16/09/2012	17/09/2012	18/09/2012	19/09/2012	20/09/2012	21/09/2012
Sev. 2	0	0	0	3	4	1
Sev. 3	28	21	27	202	528	153
Sev. 4	0	0	0	11	19	7
Sev. 5	0	0	0	10	10	3

Figure 5.12 – Severity Degree of Messages from the Test Corpus

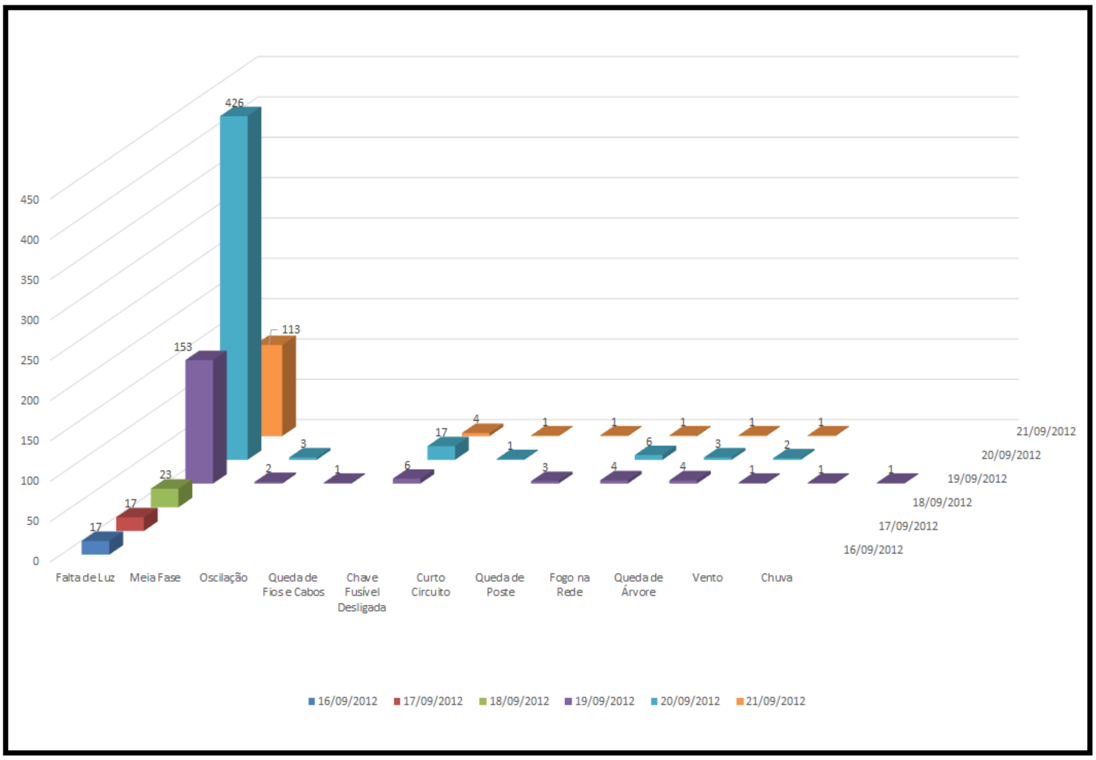


Figure 5.13 – Notification Types per day of the week - Test Corpus

to create a model of categories and notification types. This model was used in our Event Processing component, which comprises an Event Detector step based on a study over the Portuguese grammar that we conducted.

Through the evaluation of the prototype's information extraction components, we could detect flaws and improvement opportunities. We achieved good results, mainly with the Temporal Expression component and by obtaining Precision of 88% and F-measure of 71%. Although, the Event Processing component needs to include other approaches towards better event detection. Furthermore, through this chapter, we demonstrated, from the results of the study case, that the proposed IE architecture is valid.

6. CONCLUSION

Currently, more and more information are available, traveling quickly through various means of communication, being shared mainly through the Internet. To be able to make use of this information can be of great strategic interest. As the volume of information is quite extensive, it is urgent to automate this process in order to extract relevant information from texts available, while dealing with the difficulties of processing this unstructured information.

Throughout this dissertation, it was possible to further our understanding of NLP, specifically Information Extraction, its features, challenges and opportunities. Techniques such as POS Tagging, Event Detection, and Temporal Expression Recognition, among others, have the ability to increase the level of knowledge we have about the content available in text form, allowing one to perform more detailed and complex analysis of such content.

Thereby, one can structure this information, allowing the use of statistical approaches over data. The use of large textual corpora is fundamental to understand language processing, so one can strategically exploit this knowledge, even in other bases.

6.1 Contributions

Once we studied the fundamentals of the Information Extraction area and researched related work, focusing mainly on short messages application, we could understand their relevance. IE is an important research area, which is able to contribute to linguistic researches, to understand the uses and appropriations of the language, but also to assist the society with relevant information, for example, in emergency situations. During emergencies, any piece of information can help services provision. For that matter, SMS messages can be an important source of valuable information, as it is one of the most widely used means of communication.

However, users of this service usually write messages in a proper language containing abbreviations, slangs and misspellings, which hamper their processing and the extraction of useful information in their context of operation. In addition, there are many development opportunities that, to the present, have not been explored, especially in the case of the Portuguese language. Therefore, in this dissertation, we presented a proposal for an architecture to extract information from Portuguese written SMS messages under emergency circumstances.

Seizing this opportunity, we also proposed a SMS corpus building process. As, to our knowledge, there are no public SMS Corpora available in Portuguese, we expect that this process may favor other applications in this language. This process by itself is of great importance in the area as most NLP researches focus on English, and there are still

few fundamental studies and lack of tools for processing texts in Portuguese [4, 50]. This process is also dynamic, as it can be used to build corpora of tweets as well, considering the similarities between these short text messages.

The proposed IE architecture comprises a Linguistic Processing Component, which prepares the messages for Information Extraction; a Temporal Processing component, responsible for identifying and tagging existing temporal information within messages; an Event Processing component, which detects and classifies events found in the messages according to a list of domain-related categories; and an Information Fusion component that interprets information and displays them in a graph.

In Chapter 5, we detailed a case study conducted to validate the architecture. We were able to collect and filter, in an easy manner, text messages produced by clients, so that these messages can be delivered to a temporal information and event extraction application. During this study, we also built BraCorpSMS - a corpus of SMS messages in XML format already anonymized, containing an identifier and the message's delivery date.

From the extracted information, we could generate two charts to observe the messages' flow during emergencies. We were able to visualize the severity growth trend during the observed period, which could be helpful to anticipate emergencies in order to enhance services provision. These results show the applicability of the system whilst attending the purpose of the architecture.

Since we did not find similar work to compare to ours, we evaluated the Information Extraction components. Therefore, we invited three judges with domain knowledge to contribute regarding our tagging model and to use it to build a gold standard. Results obtained show that the architecture is valid and can be adapted according to the context.

IE systems built on the proposed IE architecture, like our prototype, could be adapted to attend other electric utility companies. The IE architecture could also be implemented to address other lines of business. In addition, one can use this architecture to process messages in different languages, besides Portuguese, as the Linguistic Processing component must be adapted to the messages' domain language.

6.2 Future Work

As future work, we consider including a preprocessing step on the SMS corpus building process. As data quality is important to achieve better results, it may be important to consider which methods and techniques can facilitate collecting better data for the Information Extraction process.

As for the architecture, we intend to continue our research, revising the Case Study results, and refining the model according to other approaches. One possible envisaged fea-

ture is the addition of geographic information, which can be of great importance for services provision. Moreover, the extracted temporal information may be more explored, in order to observe durations of events and their relations to notifications sent from different clients.

REFERENCES

[1] Accorsi, P.; Patel, N.; Lopez, C.; Panckhurst, R.; Roche, M. “Seek & Hide: Anonymising a French SMS Corpus Using Natural Language Processing Techniques”, *Lingvisticæ Investigationes*, vol. 35–2, 2012, pp. 163–180.

[2] Bell, E.; McGrath, L.; Gregory, M. “Verb-triggered Event Detection and Classification”. In: The Pacific Northwest Regional NLP Workshop 2010 (NW-NLP 2010), 2010.

[3] Bernicot, J.; Volckaert-Legrier, O.; Goumi, A.; Bert-Erboul, A. “Forms and Functions of SMS Messages: A Study of Variations in a Corpus Written by Adolescents”, *Journal of Pragmatics*, vol. 44–12, 2012, pp. 1701–1715.

[4] Branco, A.; Mendes, A.; Pereira, S.; Henriques, P.; Meinedo, H.; Trancoso, I.; Quaresma, P.; Strube de Lima, V. L.; Bacelar, F. “The Portuguese Language in the Digital Age / A Língua Portuguesa na Era Digital”. Springer, 2012.

[5] Carletta, J. “Assessing Agreement on Classification Tasks: The Kappa Statistic”, *Computational Linguistics*, vol. 22–2, 1996, pp. 249–254.

[6] Chen, T.; Kan, M.-Y. “Creating a Live, Public Short Message Service Corpus: The NUS SMS Corpus.”, *Language Resources and Evaluation*, vol. 47–2, 2013, pp. 299–335.

[7] Clark, A.; Fox, C.; Lappin, S. “The Handbook of Computational Linguistics and Natural Language Processing”. John Wiley & Sons, 2010, vol. 57.

[8] Corvey, W. J.; Verma, S.; Vieweg, S.; Palmer, M.; Martin, J. H. “Foundations of a Multilayer Annotation Framework for Twitter Communications During Crisis Events.” In: Proceedings of the 8^{th} International Conference on Language Resources and Evaluation Conference (LREC), 2012, pp. 21–27.

[9] Cowie, J.; Lehnert, W. “Information Extraction”, *Communications of the ACM*, vol. 39–1, 1996, pp. 80–91.

[10] da Silva, B. C. D.; Montilha, G.; Rino, L. H. M.; Specia, L.; Nunes, M. d. G. V.; de Oliveira Jr, O. N.; Martins, R. T.; Pardo, T. A. S. “Introdução ao Processamento das Línguas Naturais e Algumas Aplicações”, *Série de Relatórios do Núcleo Interinstitucional de Lingüística Computacional*, vol. 3, 2007.

[11] Dai, Y.; Kakkonen, T.; Sutinen, E. “SoMEST: a Model for Detecting Competitive Intelligence from Social Media”. In: Proceedings of the 15^{th} International Academic MindTrek Conference: Envisioning Future Media Environments, 2011, pp. 241–248.

[12] Dias, R.; Fonseca, M. J. “MuVis: An Application for Interactive Exploration of Large Music Collections”. In: Proceedings of the International Conference on Multimedia, 2010, pp. 1043–1046.

[13] Freitas, C.; Rocha, P.; Bick, E. “Floresta Sintá(c)tica: Bigger, Thicker and Easier”. In: Proceedings of the Processing of the Portuguese Language, 8^{th} International Conference (PROPOR), 2008, pp. 216—219.

[14] Gonzalez, Z. M. G. “Lingüística de Corpus na Análise do Internetês”, Master’s Thesis, Pontifical Catholic University of São Paulo, 2007.

[15] Goulart, R. R. V. “Um Modelo Híbrido para o WSD em Biomedicina”, Ph.D. Thesis, Pontifical Catholic University of Rio Grande do Sul, 2013.

[16] Han, B.; Cook, P.; Baldwin, T. “Lexical Normalization for Social Media Text”, *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 4–1, 2013, pp. 1–27.

[17] Ingersoll, G. S.; Morton, T. S.; Farris, A. L. “Taming Text: How to Find, Organize, and Manipulate it”. Manning Publications Co., 2013.

[18] Jackson, P.; Moulinier, I. “Natural Language Processing for online applications: Text retrieval, extraction and categorization”. John Benjamins Publishing, 2007, vol. 5.

[19] Jiang, J. “Information extraction from text”. In: Aggarwal, C. C. and Zhai, C., editors, Mining Text Data, 2012, pp. 11–41.

[20] Jurafsky, D.; Martin, J. H. “Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.” Prentice Hall, 2000.

[21] Kaufmann, M.; Kalita, J. “Syntactic Normalization of Twitter Messages”. In: International Conference on Natural Language Processing (ICON 2010), 2010.

[22] Kevers, L. “Accès Sémantique Aux Bases de Données Documentaires. Techniques Symboliques de Traitement Automatique du Langage pour l’Indexation Thématique et l’Extraction d’Informations Temporelles”, Ph.D. Thesis, Université Catholique de Louvain, 2011.

[23] Kobus, C.; Yvon, F.; Damnati, G. “Normalizing SMS: Are Two Metaphors Better Than One?” In: Proceedings of the 22^{nd} International Conference on Computational Linguistics, 2008, pp. 441–448.

[24] Li, C.; Sun, A.; Datta, A. “Twevent: Segment-based Event Detection from Tweets”. In: Proceedings of the 21^{st} ACM International Conference on Information and Knowledge Management, 2012, pp. 155–164.

[25] Liu, F.; Weng, F.; Jiang, X. “A Broad-coverage Normalization System for Social Media Language”. In: Proceedings of the 50^{th} Annual Meeting of the Association for Computational Linguistics, 2012, pp. 1035–1044.

[26] Lopez, C.; Bestandji, R.; Roche, M.; Panckhurst, R. “Towards Electronic SMS Dictionary Construction: An Alignment-based Approach”. In: Proceedings of the 9^{th} International Conference on Language Resources and Evaluation Conference (LREC), 2014, pp. 2833–2838.

[27] Manning, C. D.; Schütze, H. “Foundations of Statistical Natural Language Processing”. MIT Press, 1999.

[28] Mazur, P.; Dale, R. “A Rule Based Approach to Temporal Expression Tagging”. In: Proceedings of the International Multiconference on Computer Science and Information Technology, 2007, pp. 293–303.

[29] McMinn, A. J.; Moshfeghi, Y.; Jose, J. M. “Building a Large-scale Corpus for Evaluating Event Detection on Twitter”. In: Proceedings of the 22^{nd} ACM International Conference on Information & Knowledge Management, 2013, pp. 409–418.

[30] Melero, M.; Costa-Jussà, M. R.; Domingo, J.; Marquina, M.; Quixal, M. “Holaaa!! Writin Like u Talk is Kewl But Kinda Hard 4 NLP.” In: Proceedings of the 8^{th} International Conference on Language Resources and Evaluation Conference (LREC), 2012, pp. 3794–3800.

[31] Miner, G.; Elder, John, I.; Hill, T.; Nisbet, R.; Delen, D. “Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications”. Burlington, MA: Elsevier, 2012.

[32] Mitkov, R. “The Oxford handbook of computational linguistics”. Oxford University Press, 2003.

[33] Ning, Z.; Yi, J. S.; Palakal, M. J.; McDaniel, A. “OncoViz: A User-centric Mining and Visualization Tool for Cancer-related Literature”. In: Proceedings of the 2010 ACM Symposium on Applied Computing, 2010, pp. 1827–1828.

[34] Oliva, J.; Serrano, J. I.; del Castillo, M. D.; Igesias, Á. “A SMS Normalization System Integrating Multiple Grammatical Resources”, *Natural Language Engineering*, vol. 19–01, 2013, pp. 121–141.

[35] Panckhurst, R.; Détrie, C.; Lopez, C.; Moïse, C.; Roche, M.; Verine, B.; et al.. “Sud4science, de l’Acquisition d’un Grand Corpus de SMS en Français à l’Analyse de l’écriture SMS”, *Épistémè–revue Internationale de Sciences Sociales Appliquées, 9: Des Usages Numériques Aux Pratiques Scripturales Électroniques*, 2013.

[36] Parikh, R.; Karlapalem, K. “ET: Events from Tweets”. In: Proceedings of the 22^{nd} International Conference on World Wide Web Companion, 2013, pp. 613–620.

[37] Parimala, R.; Nallaswamy, R. “A Study on Analysis of SMS Classification Using Document Frequency Threshold”, *International Journal of Information Engineering and Electronic Business (IJIEEB)*, vol. 4–1, 2012, pp. 44–50.

[38] Perini, M. A. “Gramática do português brasileiro”. Parábola Ed., 2010.

[39] Popescu, A.-M.; Pennacchiotti, M.; Paranjpe, D. “Extracting Events and Event Descriptions from Twitter”. In: Proceedings of the 20^{th} International Conference Companion on World Wide Web, 2011, pp. 105–106.

[40] Pustejovsky, J.; Stubbs, A. “Natural Language Annotation for Machine Learning”. O’Reilly Media, Inc., 2012.

[41] Ritter, A.; Etzioni, O.; Clark, S.; et al.. “Open Domain Event Extraction from Twitter”. In: Proceedings of the 18^{th} ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2012, pp. 1104–1112.

[42] Russell, M. A. “Mining the Social Web: Data Mining Facebook, Twitter, LinkedIn, Google+, GitHub, and More”. O’Reilly Media, Inc., 2013.

[43] Santos, D.; Cardoso, N.; Seco, N.; Vilela, R. “Breve Introdução ao HAREM”. In: HAREM, a Primeira Avaliação Conjunta de Sistemas de Reconhecimento de Entidades Mencionadas para Português: Documentação e Actas do Encontro, Linguateca, 2007.

[44] Santos, D.; Freitas, C.; Oliveira, H. G.; Carvalho, P. “Second HAREM: New Challenges and Old Wisdom”. In: Proceedings of the Processing of the Portuguese Language, 8^{th} International Conference (PROPOR), 2008, pp. 212–215.

[45] Sardinha, T. B. “Lingüística de Corpus”. Editora Manole Ltda., 2004.

[46] Seon, C.-N.; Yoo, J.; Kim, H.; Kim, J.-H.; Seo, J. “Lightweight Named Entity Extraction for Korean Short Message Service Text”, *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 5–3, 2011, pp. 560–574.

[47] Spasić, I.; Sarafraz, F.; Keane, J. A.; Nenadić, G. “Medication Information Extraction with Linguistic Pattern Matching and Semantic Rules”, *Journal of the American Medical Informatics Association*, vol. 17–5, 2010, pp. 532–535.

[48] Sridhar, V. K. R.; Chen, J.; Bangalore, S.; Shacham, R. “A Framework for Translating SMS Messages”. In: Proceedings of COLING 2014, the 25^{th} International Conference on Computational Linguistics: Technical Papers, Dublin, Ireland, 2014, pp. 974–983.

[49] Stein, C.; Devore, R.; Wojcik, B. “Calculation of the Kappa Statistic for Inter-rater Reliability: The Case Where Raters Can Select Multiple Responses from a Large Number of Categories”. In: Proceedings of the SAS® Users Group International Conference. Cary, NC, USA: SAS Institute Inc, 2005.

[50] Strube de Lima, V. L.; Nunes, M. d. G. V.; Vieira, R. “Desafios do Processamento das Línguas Naturais”. In: Anais do XXVII Congresso da SBC, 2007, pp. 2202–2216.

[51] Tagg, C. “A Corpus Linguistics Study of SMS Text Messaging”, Ph.D. Thesis, The University of Birmingham, 2009.

[52] Tagnin, S. E. O.; Vale, O. A. “Avanços da Linguística de Corpus no Brasil”. Editora Humanitas, 2008.

[53] Thelwall, M.; Buckley, K.; Paltoglou, G. “Sentiment in Twitter Events”, *Journal of the American Society for Information Science and Technology*, vol. 62–2, 2011, pp. 406–418.

[54] TimeML Working Group. “Guidelines for temporal expression annotation for english for tempeval 2010”, 2009.

[55] Treurniet, M.; De Clercq, O.; van den Heuvel, H.; Oostdijk, N. “Collection of a Corpus of Dutch SMS”. In: Proceedings of the 8^{th} International Conference on Language Resources and Evaluation Conference (LREC), 2012, pp. 2268–2273.

[56] Wang, X.; Zhu, F.; Jiang, J.; Li, S. “Real Time Event Detection in Twitter”. In: Proceedings of the 13^{th} International Conference (WAIM 2012), 2012, pp. 502–513.

[57] Wang, Y.; Zhu, M.; Qu, L.; Spaniol, M.; Weikum, G. “Timely YAGO: Harvesting, Querying, and Visualizing Temporal Knowledge from Wikipedia”. In: Proceedings of the 13^{th} International Conference on Extending Database Technology, 2010, pp. 697–700.

[58] Weiser, S.; Cougnon, L.-A.; Watrin, P. “Temporal Expressions Extraction in SMS Messages”. In: Proceedings of the Workshop on Information Extraction and Knowledge Acquisition, Hissar, Bulgaria, September, 2011, pp. 41–44.

[59] Williams, J.; Katz, G. “A New Twitter Verb Lexicon for Natural Language Processing”. In: Proceedings of the 8^{th} International Conference on Language Resources and Evaluation Conference (LREC), 2012, pp. 293–298.

APPENDIX A – TIMEX2 TAGSET

The TIMEX standard is built on the ISO 8601 standard, where temporal expressions are annotated using XML tags. TIMEX2 tags can have one or more attributes, which are:

Attribute	Description	Examples
VAL	Normalization of expressions of calendrical/clock times and durations. Uses extended ISO 8601 scheme to capture broader ranges of time	1986, SU (summer), NI (night), P1D (one day period), PAST_REF
MOD	Normalization of modifiers, such as “no later than”, “late”. Appears in combination with VAL, indicating "point" expressions, "duration" expressions or points and durations.	BEFORE, AFTER, LESS_THAN, MORE_THAN, START, APPROX
SET	Designates set-denoting expressions. Almost always appears in combination with VAL.	Only value is YES.
GRANULARITY	Normalization of granularity of set-denoting expression. "Every December 31" has granularity "G1D” (one day). Always appears in combination with SET.	G1D, G.5M
PERIODICITY	Normalization of periodicity of set-denoting expression. Ex.: “every December 31” has periodicity value of “F1Y” (frequency of one year). Always appears in combination with SET, and only when the set denotes a regularly recurring set	F1Y, F.5M
ANCHOR_VAL	Normalization of reference calendar/clock time for interpreting a vague or partially anchored expression such as “now”, “the past few years”. Appears in combination with VAL for such expressions. The reference time can be the “narrative” time or the “document” time.	1986-03, 1997-12-26T08:26
ANCHOR_DIR	Normalization of directionality. Appears in combination with ANCHOR_VAL. For example, “now” has anchor directionality of “AS_OF”, “the past few years” has anchor directionality of “BEFORE”.	AS_OF, BEFORE
NON_SPECIFIC	Designates a generic, indefinite or non-referential time expression Ex.: “I love December”/“He voted on a Tuesday”/“It’s a sunny day”	
COMMENT	Contains any comments that may seem fit.	

APPENDIX B – POS TAGGER EVALUATION

This appendix details the evaluation results of NLTK's POS Taggers trained over MacMorpho and Floresta. These tests were conducted while prototyping the Information Extraction system.

#	Message	Tokenized Sentence	Mac_Morpho	Hits	Misses	Floresta	Hits	Misses
1	LUZ NUMERO DA INSTALACAO #IDNUM AINDA ESTAMOS SEM LUZ	luz numero da instalacao ainda estamos sem luz	(luz, N) (numero, None) (da, NPROP) (instalacao, None) (ainda, ADV) (estamos, V) (sem, PREP) (luz, N)	5	3	(luz, n) (numero, None) (da, None) (instalacao, None) (ainda, adv) (estamos, v-fin) (sem, prp) (luz, n)	5	3
2	ESTAMOS SEM LUZ DESDE INICIO DA NOITE: RUA PROFESSOR DOUTOR MILTON GUERREIRO, 226 - CEP 90.850-350. PORTO ALEGRE.	estamos sem luz desde inicio da noite rua professor doutor milton guerreiro 226 cep 90850350	(estamos, V) (sem, PREP) (luz, N) (desde, PREP) (inicio, V) (da, NPROP) (noite, NPROP) (rua, NPROP) (professor, NPROP) (doutor, N) (milton, NPROP) (guerreiro, NPROP) (226, NUM) (cep, N) (90850350, None)	10	5	(estamos, v-fin) (sem, prp) (luz, n) (desde, prp) (inicio, None) (da, None) (noite, n) (rua, n) (professor, n) (doutor, n) (milton, None) (guerreiro, n) (226, num) (cep, n) (90850350, None)	11	4
		porto alegre	(porto, NPROP) (alegre, NPROP)	2	0	(porto, prop) (alegre, adj)	1	1
3	LUZ ESTAMOS SEM LUZ DEUS DE ONTEM #IDNUM	luz estamos sem luz deus de ontem	(luz, N) (estamos, VAUX) (sem, PREP) (luz, N) (deus, N) (de, PREP) (ontem, ADV)	5	2	(luz, n) (estamos, v-fin) (sem, prp) (luz, n) (deus, prop) (de, prp) (ontem, adv)	5	2
4	ESTAMOS SEM LUZ DEIS DE ONTEM DE TARDE CAIU PAU DE ACACIA E FIO ESTA CAIDO NO INDAIA.	estamos sem luz deis de ontem de tarde caiu pau de acacia fio caido indaia	(estamos, V) (sem, PREP) (luz, N) (deis, None) (de, PREP) (ontem, ADV) (de,) (tarde, N) (caiu, V) (pau, NPROP) (de, NPROP) (acacia, None) (fio, N) (caido, None) (indaia, None)	8	7	(estamos, v-fin) (sem, prp) (luz, n) (deis, None) (de, prp) (ontem, adv) (de, prp) (tarde, n) (caiu, v-fin) (pau, n) (de, prp) (acacia, None) (fio, n) (caido, None) (indaia, None)	10	5
5	LUZ #IDNUM DICA: ALIMENTADOR BAR NEFTON CAI FICHA	luz dica alimentador bar nefton cai ficha	(luz, N) (dica, N) (alimentador, None) (bar, N) (nefton, None) (cai, V) (ficha, N)	5	2	(luz, n) (dica, n) (alimentador, None) (bar, n) (nefton, None) (cai, v-fin) (ficha, n)	5	2
6	LUZ#IDNUM REDE EM MEIA FASE	luz#idnum rede meia fase	(luz#idnum, None) (rede, N) (meia, N) (fase, N)	2	2	(luz#idnum, None) (rede, n) (meia, n) (fase, n)	2	2
7	LUZ.#IDNUM FALTANDO DES DE 4 HORAS DA MANHA.	luz faltando des de 4 horas da manha	(luz, N) (faltando, V) (des, NPROP) (de, NPROP) (4, N|AP) (horas, N) (da, N) (manha, None)	4	4	(luz, n) (faltando, v-ger) (des, None) (de, prp) (4, num) (horas, n) (da, None) (manha, None)	5	3
8	ESTAMOS COM A LUZ EM MEIA FAZE #IDNUM	estamos com luz meia faze	(estamos, V) (com, PREP) (luz, N) (meia, N) (faze, V)	3	2	(estamos, v-fin) (com, prp) (luz, n) (meia, n) (faze, None)	4	1

#	Message	Tokenized Sentence	Mac_Morpho	Hits	Misses	Floresta	Hits	Misses
9	LUZ #IDNUM DEU UM ESTOURO E OS FIOS DO POSTE ESTAO TUDO NO CHAO...	luz deu um estouro fios do poste estao tudo chao	(luz, N) (deu, V) (um, ART) (estouro, N) (fios, N) (do, N) (poste, N) (estao, None) (tudo, PROSUB) (chao, None)	7	3	(luz, n) (deu, v-fin) (um, art) (estouro, None) (fios, n) (do, None) (poste, n) (estao, None) (tudo, pron-indp) (chao, None)	6	4
10	LUZ/RISCO.CHOQ.ENTROU,EN CURTO E CAIO A CONECXAO DOS FIOS URGENTE!LUZ#IDNUM	luz risco	(luz, N) (risco, N)	2	0	(luz, n) (risco, n)	2	0
		choq	(choq, None)	0	1	(choq, None)	0	1
		entrou curto caio conecxao dos fios urgente	(entrou, V) (curto, ADJ) (caio, NPROP) (conecxao, None) (dos, NPROP) (fios, N) (urgente, ADJ)	3	4	(entrou, v-fin) (curto, adj) (caio, prop) (conecxao, None) (dos, prop) (fios, n) (urgente, adj)	3	4
		luz	(luz, N)	1	0	(luz, n)	1	0
11	FALTA D LUZ DEVIDO AO CURTO NA RED DE ALTA TENSAO NO RINCAO DOS AMERICO. #IDNUM	falta d luz devido curto red de alta tensao rincao dos americo	(falta, V) (d,) (luz, N) (devido,) (curto, ADJ) (red, NPROP) (de, NPROP) (alta, NPROP) (tensao, None) (rincao, None) (dos, NPROP) (americo, None)	3	9	(falta, v-fin) (d, None) (luz, n) (devido, prp) (curto, adj) (red, adj) (de, prp) (alta, n) (tensao, None) (rincao, None) (dos, prop) (americo, None)	5	7
12	OLA QUANDO RETONA A LUZ NO BAIRRO SANTO ANTONIO? FALTA DESDE AS 15H?!	ola quando retona luz bairro santo antonio	(ola, None) (quando, KS) (retona, None) (luz, N) (bairro, N) (santo, NPROP) (antonio, NPROP)	5	2	(ola, None) (quando, adv) (retona, None) (luz, n) (bairro, n) (santo, None) (antonio, None)	3	4
		falta desde 15h	(falta, V) (desde, PREP) (15h, N|HOR)	3	0	(falta, v-fin) (desde, prp) (15h, n)	2	1
13	LUZ #IDNUM .JA VAO SE PASSAR 3 HORAS E CONTINUAMOS SEM LUZ NA RUA.OBRIGADO	luz	(luz, N)	1	0	(luz, n)	1	0
		vao passar 3 horas continuamos sem luz rua	(vao, None) (passar, V) (3, NUM) (horas, N) (continuamos, VAUX) (sem, PREP) (luz, N) (rua, N)	7	1	(vao, None) (passar, v-inf) (3, num) (horas, n) (continuamos, v-fin) (sem, prp) (luz, n) (rua, n)	7	1
14	ESTAMOS SEM LUZ MAIS OU MENOS MEIA HORA #IDNUM	estamos sem luz mais ou menos meia hora	(estamos, V) (sem, PREP) (luz, N) (mais, ADV) (ou, ADV) (menos, ADV) (meia, ADV) (hora, N)	7	1	(estamos, v-fin) (sem, prp) (luz, n) (mais, adv) (ou, conj-c) (menos, adv) (meia, n) (hora, n)	6	2

#	Message	Tokenized Sentence	Mac_Morpho	Hits	Misses	Floresta	Hits	Misses
15	ESTAMOS SEM LUZ A MAIS DE 3 HORAS, R:OSCAR FERREIRA 115, P.DOS MAIAS QUANDO VAMOS TER ENERGIA? JOVENIL	estamos sem luz mais de 3 horas r oscar ferreira 115	(estamos, V) (sem, PREP) (luz, N) (mais, ADV) (de, PREP) (3, NUM) (horas, N) (r, NPROP) (oscar, NPROP) (ferreira, NPROP) (115, NUM)	11	0	(estamos, v-fin) (sem, prp) (luz, n) (mais, adv) (de, prp) (3, num) (horas, n) (r, prop) (oscar, n) (ferreira, None) (115, num)	9	2
		dos maias quando vamos ter energia	(dos, NPROP) (maias, NPROP) (quando, KS) (vamos, VAUX) (ter, VAUX) (energia, N)	6	0	(dos, prop) (maias, None) (quando, adv) (vamos, v-fin) (ter, v-inf) (energia, n)	5	1
		jovenil	(jovenil, None)	0	1	(jovenil, None)	0	1
16	LUZ #IDNUM CABO CAIDO NA RUA	luz cabo caido rua	(luz, N) (cabo, N) (caido, None) (rua, N)	3	1	(luz, n) (cabo, n) (caido, None) (rua, n)	3	1
17	FALTOU LUZ NA JACIPUIA. BAIRRO GUARUJA A MAIS DE UMA HORA. TEM PREVISAO DE VOLTA?	faltou luz jacipuia	(faltou, V) (luz, N) (jacipuia, None)	2	1	(faltou, v-fin) (luz, n) (jacipuia, None)	2	1
		bairro guaruja mais de uma hora	(bairro, N) (guaruja, None) (mais, ADV) (de,) (uma, ART) (hora, N)	4	2	(bairro, n) (guaruja, None) (mais, adv) (de, prp) (uma, art) (hora, n)	5	1
		tem previsao de volta	(tem, V) (previsao, None) (de, PREP) (volta, N)	3	1	(tem, v-fin) (previsao, None) (de, prp) (volta, n)	3	1
18	LUZ INSTALACAO #IDNUM SEM LUZ HA MAIS DE 2 HORAS.	luz instalacao sem luz ha mais de 2 horas	(luz, N) (instalacao, None) (sem, PREP) (luz, N) (ha, N) (mais, ADV) (de, PREP) (2, NUM) (horas, N)	7	2	(luz, n) (instalacao, None) (sem, prp) (luz, n) (ha, n) (mais, adv) (de, prp) (2, num) (horas, n)	7	2
19	LUZ.FALTOU LUZ EM TODA A CIDADE DE ALVORADA.ANDREIA	luz faltou luz toda cidade de alvorada	(luz, N) (faltou, V) (luz, N) (toda, PROADJ) (cidade, N) (de, PREP) (alvorada, NPROP)	7	0	(luz, n) (faltou, v-fin) (luz, n) (toda, pron-det) (cidade, n) (de, prp) (alvorada, None)	6	1
		andreia	(andreia, None)	0	1	(andreia, None)	0	1
20	LUZ #IDNUM CAIO ARVORE NA RUA NOS FIOS.	luz caio arvore rua fios	(luz, N) (caio, NPROP) (arvore, None) (rua, N) (fios, N)	3	2	(luz, n) (caio, prop) (arvore, None) (rua, n) (fios, n)	3	2
21	LUZ INSTALACAO NUMERO #IDNUM . ACHO QUE QUEIMOU FUZIL DO TRANSFORMADOR. JORGE DOS SANTOS E	luz instalacao numero	(luz, N) (instalacao, None) (numero, None)	1	2	(luz, n) (instalacao, None) (numero, None)	1	2

#	Message	Tokenized Sentence	Mac_Morpho	Hits	Misses	Floresta	Hits	Misses
	SILVA. CERRO DOS CAMARGOS CERRO GRANDE DO SUL.	acho queimou fuzil do transformador	(acho, V) (queimou, V) (fuzil, N) (do, KS) (transformador, ADJ)	3	2	(acho, v-fin) (queimou, None) (fuzil, n) (do, None) (transformador, None)	2	3
		jorge dos santos silva	(jorge, NPROP) (dos, NPROP) (santos, NPROP) (silva, NPROP)	4	0	(jorge, None) (dos, prop) (santos, prop) (silva, prop)	3	1
		cerro dos camargos cerro grande do sul	(cerro, None) (dos, NPROP) (camargos, NPROP) (cerro, None) (grande, ADJ) (do, KS) (sul, NPROP)	4	3	(cerro, None) (dos, prop) (camargos, None) (cerro, None) (grande, adj) (do, None) (sul, n)	3	4
22	JA FAZ QUASE UMA HORA Q ESTAMOS SEM LUZ AQUI EM ALVORADA BAIRRO JARDIM PORTO ALEGRE PROXIMO A TIRADENTES	faz quase uma hora estamos sem luz aqui alvorada bairro jardim porto alegre proximo tiradentes	(faz, V) (quase, ADV) (uma, ART) (hora, N) (estamos, VAUX) (sem, PREP) (luz, N) (aqui, ADV) (alvorada, NPROP) (bairro, N) (jardim, NPROP) (porto, NPROP) (alegre, NPROP) (proximo, None) (tiradentes, NPROP)	14	1	(faz, v-fin) (quase, adv) (uma, art) (hora, n) (estamos, v-fin) (sem, prp) (luz, n) (aqui, adv) (alvorada, None) (bairro, n) (jardim, n) (porto, prop) (alegre, adj) (proximo, None) (tiradentes, None)	11	4
23	FALTOU LUZ DESDE ONTEM A TARDE #IDNUM	faltou luz desde ontem tarde	(faltou, V) (luz, N) (desde, PREP) (ontem, ADV) (tarde, ADV)	5	0	(faltou, v-fin) (luz, n) (desde, prp) (ontem, adv) (tarde, adv)	5	0
24	AUSENCIA DE ENERGIA ESTRADA FAXINAL QUEIMADO. FIO DE ALTA TENCAO SOBRE O CHAO INSTALACAO N #IDNUM URGENTE	ausencia de energia estrada faxinal queimado	(ausencia, None) (de, PREP) (energia, N) (estrada, N) (faxinal, None) (queimado, PCP)	3	3	(ausencia, None) (de, prp) (energia, n) (estrada, n) (faxinal, None) (queimado, v-pcp)	3	3
		fio de alta tencao sobre chao instalacao urgente	(fio, N) (de, PREP) (alta, N) (tencao, None) (sobre, PREP) (chao, None) (instalacao, None) (urgente, ADJ)	5	3	(fio, n) (de, prp) (alta, n) (tencao, None) (sobre, prp) (chao, None) (instalacao, None) (urgente, adj)	5	3
25	NAO TEM LUZ, DESDE CEDO .#IDNUM	nao tem luz desde cedo	(nao, ADV) (tem, V) (luz, N) (desde, PREP) (cedo, ADV)	5	0	(nao, None) (tem, v-fin) (luz, n) (desde, prp) (cedo, adv)	4	1
26	JA FAZ 2HS SEM LUZ.CONTRATO #IDNUM	faz 2hs sem luz contrato	(faz, V) (2hs, None) (sem, PREP) (luz, N) (contrato, N)	4	1	(faz, v-fin) (2hs, None) (sem, prp) (luz, n) (contrato, n)	4	1
27	LUZ#IDNUM CONTINUAMOS SEM LUZ CASAS TEN OUTRAS MEIA FASE	luz continuamos sem luz casas ten outras meia fase	(luz, N) (continuamos, VAUX) (sem, PREP) (luz, N) (casas, N) (ten, N|EST) (outras, PROADJ) (meia, N) (fase, N)	7	2	(luz, n) (continuamos, v-fin) (sem, prp) (luz, n) (casas, n) (ten, None) (outras, pron-det) (meia, n) (fase, n)	7	2
28	LUZ #IDNUM MAS QUAL O PROBLEMA??? JA	luz mas qual problema	(luz, N) (mas, KC) (qual, PRO-KS) (problema, N)	4	0	(luz, n) (mas, conj-c) (qual, pron-det) (problema, n)	4	0

#	Message	Tokenized Sentence	Mac_Morpho	Hits	Misses	Floresta	Hits	Misses
	ESTAMOS MAIS DE 6 HORAS SEM LUZ!	estamos mais de 6 horas sem luz	(estamos, V) (mais, ADV) (de, PREP) (6, NUM) (horas, N) (sem, PREP) (luz, N)	7	0	(estamos, v-fin) (mais, adv) (de, prp) (6, num) (horas, n) (sem, prp) (luz, n)	7	0
29	ESTAMOS CEM LUZ EM SANTO ANTONIO DA PATRULHA NA RUA REPUBLICA ARGENTINA DES DAS TRES HORAS SOLICITAMIOS A REVISAO PRECISAMOS DE LUZ.OBRIGADO	estamos cem luz santo antonio da patrulha rua republica argentina des das tres horas solicitamios revisao precisamos de luz	(estamos, V) (cem, NUM) (luz, N) (santo, NPROP) (antonio, NPROP) (da, NPROP) (patrulha, N) (rua, N) (republica, None) (argentina, NPROP) (des, NPROP) (das, NPROP) (tres, NPROP) (horas, NPROP) (solicitamios, None) (revisao, None) (precisamos, V) (de, PREP) (luz, N)	11	8	(estamos, v-fin) (cem, num) (luz, n) (santo, None) (antonio, None) (da, None) (patrulha, n) (rua, n) (republica, None) (argentina, prop) (des, None) (das, None) (tres, prop) (horas, n) (solicitamios, None) (revisao, None) (precisamos, v-fin) (de, prp) (luz, n)	9	10
30	LUZ #IDNUM CAIU O FUZIL DO TRASFORMADOR, O FREEZER TA CHEIO DE CARNE E TEM GENTE DE IDADE E CRIANCA PEQUENA.	luz caiu fuzil do trasformador freezer ta cheio de carne tem gente de idade crianca pequena	(luz, N) (caiu, V) (fuzil, N) (do, KS) (trasformador, None) (freezer, N) (ta, None) (cheio, ADJ) (de,) (carne, N) (tem, V) (gente, N) (de, PREP) (idade, N) (crianca, None) (pequena, ADJ)	11	5	(luz, n) (caiu, v-fin) (fuzil, n) (do, None) (trasformador, None) (freezer, None) (ta, None) (cheio, adj) (de, prp) (carne, n) (tem, v-fin) (gente, n) (de, prp) (idade, n) (crianca, None) (pequena, adj)	11	5
31	BOA NOITE ESTAMOS SEM LUZ A MAIS DE UMA HORA SABE ME ENFORMAR SE O PROBLEMA JA ESTA SENDO RESOLVIDO OBRIGADO	boa noite estamos sem luz mais de uma hora sabe me enformar problema sendo resolvido	(boa, ADJ) (noite, N) (estamos, VAUX) (sem, PREP) (luz, N) (mais, ADV) (de, PREP) (uma, ART) (hora, N) (sabe, V) (me, PROPESS) (enformar, None) (problema, N) (sendo, VAUX) (resolvido, PCP)	14	1	(boa, adj) (noite, n) (estamos, v-fin) (sem, prp) (luz, n) (mais, adv) (de, prp) (uma, art) (hora, n) (sabe, v-fin) (me, pron-pers) (enformar, None) (problema, n) (sendo, v-ger) (resolvido, v-pcp)	14	1
32	NUMERO #IDNUM ESTAMOS SEM LUZ A DUAS HORAS QUAL O PROBEMA PARA TANTA DEMORA	numero estamos sem luz duas horas qual probema tanta demora	(numero, None) (estamos, V) (sem, PREP) (luz, N) (duas, NUM) (horas, N) (qual, PRO-KS) (probema, None) (tanta, PROADJ) (demora, N)	8	2	(numero, None) (estamos, v-fin) (sem, prp) (luz, n) (duas, num) (horas, n) (qual, pron-det) (probema, None) (tanta, pron-det) (demora, None)	7	3
33	A 2 DIAS ESTAMOS SEM LUZ. LUZ #IDNUM	2 dias estamos sem luz luz	(2, N) (dias, N) (estamos, VAUX) (sem, PREP) (luz, N) (luz, NPROP)	4	2	(2, num) (dias, n) (estamos, v-fin) (sem, prp) (luz, n) (luz, n)	6	0

#	Message	Tokenized Sentence	Mac_Morpho	Hits	Misses	Floresta	Hits	Misses
34	LUZ #IDNUM A QUASE 3 HORAS NAO A LUZ SEMPRE QUE CHOVE UM POUCO OU VENTA ESSA PORCARIA FALTA	luz quase 3 horas nao luz sempre chove um pouco ou venta porcaria falta	(luz, N) (quase, ADV) (3, NUM) (horas, N) (nao, ADV) (luz, N) (sempre, ADV) (chove, V) (um, ART) (pouco, PROADJ) (ou, KC) (venta, None) (porcaria, N) (falta, V)	12	2	(luz, n) (quase, adv) (3, num) (horas, n) (nao, None) (luz, n) (sempre, adv) (chove, v-fin) (um, art) (pouco, adv) (ou, conj-c) (venta, None) (porcaria, None) (falta, n)	11	3
35	LUZ #IDNUM SEM LUZ DESDE 16H40M.	luz sem luz desde 16h40m	(luz, N) (sem, PREP) (luz, N) (desde, PREP) (16h40m, None)	4	1	(luz, n) (sem, prp) (luz, n) (desde, prp) (16h40m, None)	4	1
36	LUZ #IDNUM JA FAZ 24 HORAS QUE FOI EFETUADA A PRIMEIRA CHAMADA!	luz faz 24 horas foi efetuada primeira chamada	(luz, N) (faz, V) (24, NUM) (horas, N) (foi, V) (efetuada, PCP) (primeira, ADJ) (chamada, PCP)	8	0	(luz, n) (faz, v-fin) (24, num) (horas, n) (foi, v-fin) (efetuada, None) (primeira, adj) (chamada, n)	7	1
37	LUZ #IDNUM URGENTE JA FAZ 4 HORAS SEM LUZ,POR FAVOR	luz urgente faz 4 horas sem luz favor	(luz, N) (urgente, ADJ) (faz, V) (4, NUM) (horas, N) (sem, PREP) (luz, N) (favor,)	7	1	(luz, n) (urgente, adj) (faz, v-fin) (4, num) (horas, n) (sem, prp) (luz, n) (favor, n)	8	0
38	FALTAS INTERMITENTES DE LUZ DESDE AS 17H. NUMERO DE INSTALACAO #IDNUM	faltas intermitentes de luz desde 17h	(faltas, N) (intermitentes, ADJ) (de,) (luz, N) (desde, PREP) (17h, N|HOR)	5	1	(faltas, n) (intermitentes, None) (de, prp) (luz, n) (desde, prp) (17h, n)	4	2
		numero de instalacao	(numero, None) (de, PREP) (instalacao, None)	1	2	(numero, None) (de, prp) (instalacao, None)	1	2
39	ESTAMOS SEM LUZ DESDE AS 18H. INSTALACAO N #IDNUM .	estamos sem luz desde 18h	(estamos, V) (sem, PREP) (luz, N) (desde, PREP) (18h, N|HOR)	5	0	(estamos, v-fin) (sem, prp) (luz, n) (desde, prp) (18h, n)	4	1
		instalacao	(instalacao, None)	0	1	(instalacao, None)	0	1
40	JA ESTAMOS SEM LUZ DESDE AS 16 HS.ISSO NUM DOMINGO DE CHUVA E HORRIVEL.	estamos sem luz desde 16 hs isso num domingo de chuva horrivel	(estamos, V) (sem, PREP) (luz, N) (desde, PREP) (16, NUM) (hs, None) (isso, PROSUB) (num, NPROP) (domingo, NPROP) (de, NPROP) (chuva, NPROP) (horrivel, None)	6	6	(estamos, v-fin) (sem, prp) (luz, n) (desde, prp) (16, num) (hs, None) (isso, pron-indp) (num, None) (domingo, n) (de, prp) (chuva, n) (horrivel, None)	9	3
41	ESTAMOS SEM LUZ A MAIS DE TRES HORA E MEIA NR. #IDNUM	estamos sem luz mais de tres hora meia nr	(estamos, V) (sem, PREP) (luz, N) (mais, ADV) (de, PREP) (tres, NPROP) (hora, NPROP) (meia, NPROP) (nr, NPROP)	5	4	(estamos, v-fin) (sem, prp) (luz, n) (mais, adv) (de, prp) (tres, prop) (hora, n) (meia, n) (nr, None)	6	3
42	LUZ #IDNUM ,A ENERGIA VEIO E VOLTOU VARIAS VZES.	luz energia veio voltou varias vzes	(luz, N) (energia, N) (veio, V) (voltou, V) (varias, None) (vzes, None)	4	2	(luz, n) (energia, n) (veio, v-fin) (voltou, v-fin) (varias, None) (vzes, None)	4	2

#	Message	Tokenized Sentence	Mac_Morpho	Hits	Misses	Floresta	Hits	Misses
43	LUZ,FUI ATE A RUA QUE FALTOU LUZ E NAO VIMOS MENHUMA MOVIMENTACAO DA CEEE ESTAMOS DES DO MEIO DIA SEM LUZ. MINHA ESTALACAO E #IDNUM	luz fui ate rua faltou luz nao vimos menhuma movimentacao da ceee estamos des do meio dia sem luz minha estalacao	(luz, N) (fui, VAUX) (ate, None) (rua, N) (faltou, V) (luz, N) (nao, ADV) (vimos, V) (menhuma, None) (movimentacao, None) (da, NPROP) (ceee, NPROP) (estamos, VAUX) (des, NPROP) (do, NPROP) (meio, NPROP) (dia, NPROP) (sem, NPROP) (luz, NPROP) (minha, NPROP) (estalacao, None)	9	12	(luz, n) (fui, v-fin) (ate, None) (rua, n) (faltou, v-fin) (luz, n) (nao, None) (vimos, v-fin) (menhuma, None) (movimentacao, None) (da, None) (ceee, None) (estamos, v-fin) (des, None) (do, None) (meio, n) (dia, n) (sem, prp) (luz, n) (minha, pron-det) (estalacao, None)	10	11
44	ESTAMOS SEM LUZ DESDE ONTEM AS 19 HORAS.	estamos sem luz desde ontem 19 horas	(estamos, V) (sem, PREP) (luz, N) (desde, PREP) (ontem, ADV) (19, NUM) (horas, N)	7	0	(estamos, v-fin) (sem, prp) (luz, n) (desde, prp) (ontem, adv) (19, num) (horas, n)	7	0
45	LUZ,DESD SABADO,CAIU A CHAVE D TRANSFORMADOR,#IDNUM	luz desd sabado caiu chave d transformador	(luz, N) (desd, None) (sabado, None) (caiu, V) (chave, N) (d, N) (transformador, ADJ)	3	4	(luz, n) (desd, None) (sabado, None) (caiu, v-fin) (chave, n) (d, None) (transformador, None)	3	4
46	FALTA DE LUZ DAS DEZOITO HORAS DE DOMINGO #IDNUM ATE AGORA	falta de luz das dezoito horas de domingo ate agora	(falta, V) (de, PREP) (luz, N) (das, NPROP) (dezoito, NUM) (horas, N) (de, PREP) (domingo, N) (ate, None) (agora, ADV)	8	2	(falta, v-fin) (de, prp) (luz, n) (das, None) (dezoito, None) (horas, n) (de, prp) (domingo, n) (ate, None) (agora, adv)	7	3
47	FALTA DE LUZ DESDE ONTEM POR FAVOR.#IDNUM	falta de luz desde ontem favor	(falta, V) (de, PREP) (luz, N) (desde, PREP) (ontem, ADV) (favor,)	5	1	(falta, v-fin) (de, prp) (luz, n) (desde, prp) (ontem, adv) (favor, n)	6	0
48	FALTA D LUZ A 22 HORAS JA #IDNUM	falta d luz 22 horas	(falta, V) (d,) (luz, N) (22, N|AP) (horas, N)	3	2	(falta, v-fin) (d, None) (luz, n) (22, num) (horas, n)	4	1
49	LUZ #IDNUM MEIA FASE DESE DAS 10:00 DE ONTEM	luz meia fase dese das 10:00 de ontem	(luz, N) (meia, N) (fase, N) (dese, None) (das, NPROP) (10:00, None) (de, PREP) (ontem, ADV)	4	4	(luz, n) (meia, n) (fase, n) (dese, None) (das, None) (10:00, None) (de, prp) (ontem, adv)	4	4
50	LUZ ESTAMOS A 40 HORAS SEM LUZ #IDNUM	luz estamos 40 horas sem luz	(luz, N) (estamos, VAUX) (40, NUM) (horas, N) (sem, PREP) (luz, N)	6	0	(luz, n) (estamos, v-fin) (40, num) (horas, n) (sem, prp) (luz, n)	6	0
51	A MINHA LUZ ESTA SO NUMA FAZE #IDNUM	minha luz so numa faze	(minha, PROADJ) (luz, N) (so, None) (numa, NPROP) (faze, V)	2	3	(minha, pron-det) (luz, n) (so, None) (numa, None) (faze, None)	2	3
52	#IDNUM JA ESTAMOS A 24 HORAS SEM LUZ - TODA A RUA-.	estamos 24 horas sem luz toda rua	(estamos, V) (24, NUM) (horas, N) (sem, PREP) (luz, N) (toda, PROADJ) (rua, N)	7	0	(estamos, v-fin) (24, num) (horas, n) (sem, prp) (luz, n) (toda, pron-det) (rua, n)	6	1

#	Message	Tokenized Sentence	Mac_Morpho	Hits	Misses	Floresta	Hits	Misses
53	FALTA DE LUZ DESDE ONTEM #IDNUM	falta de luz desde ontem	(falta, V) (de, PREP) (luz, N) (desde, PREP) (ontem, ADV)	5	0	(falta, v-fin) (de, prp) (luz, n) (desde, prp) (ontem, adv)	5	0
54	LUZ #IDNUM FALTOU DESDE AS 10:30	luz faltou desde 10:30	(luz, N) (faltou, V) (desde, PREP) (10:30, None)	3	1	(luz, n) (faltou, v-fin) (desde, prp) (10:30, None)	3	1
55	FALTA LUZ DES DE ONTEN DAS 15,00 HORAS NOMERO DA INSTALACAO #IDNUM	falta luz des de onten das 15 00 horas nomero da instalacao	(falta, V) (luz, N) (des, NPROP) (de, NPROP) (onten, None) (das, NPROP) (15, NPROP) (00, N|AP) (horas, N) (nomero, None) (da, NPROP) (instalacao, None)	3	9	(falta, v-fin) (luz, n) (des, None) (de, prp) (onten, None) (das, None) (15, num) (00, None) (horas, n) (nomero, None) (da, None) (instalacao, None)	5	7
56	LUZ EM MEIA FASE RUA VEADOR PORTO ENTRE RUA SAO LUIZ E SAO MANOEL CAIU UM RAIO.SOLICITO ESPECIAL ATENDIMENTO.OBRIGADO INSTALACAO N #IDNUM .	luz meia fase rua veador porto entre rua sao luiz sao manoel caiu um raio	(luz, N) (meia, N) (fase, N) (rua, N) (veador, None) (porto, NPROP) (entre, PREP) (rua, N) (sao, None) (luiz, NPROP) (sao, None) (manoel, NPROP) (caiu, V) (um, ART) (raio, N)	11	4	(luz, n) (meia, n) (fase, n) (rua, n) (veador, None) (porto, prop) (entre, prp) (rua, n) (sao, None) (luiz, None) (sao, None) (manoel, None) (caiu, v-fin) (um, art) (raio, n)	9	6
		solicito especial atendimento	(solicito, None) (especial, ADJ) (atendimento, N)	2	1	(solicito, None) (especial, ADJ) (atendimento, N)	2	1
		instalacao	(instalacao, None)	0	1	(instalacao, None)	0	1
57	JA ESTAMOS A 28HORAS SEM LUZ	estamos 28horas sem luz	(estamos, V) (28horas, None) (sem, PREP) (luz, N)	3	1	(estamos, v-fin) (28horas, None) (sem, prp) (luz, n)	3	1
58	ESTAMOS SO NUMA FAZE LUZ #IDNUM	estamos so numa faze luz	(estamos, V) (so, None) (numa, NPROP) (faze, V) (luz, N)	2	3	(estamos, v-fin) (so, None) (numa, None) (faze, None) (luz, n)	2	3
59	ESTOU SEM LUZ A QUASE A MAIS DE UMA HORA.	estou sem luz quase mais de uma hora	(estou, V) (sem, PREP) (luz, N) (quase, ADV) (mais, ADV) (de,) (uma, ART) (hora, N)	7	1	(estou, v-fin) (sem, prp) (luz, n) (quase, adv) (mais, adv) (de, prp) (uma, art) (hora, n)	8	0
60	CURTO NA LUZ N #IDNUM ULTIMO CONTATO SERA COM A IMPRENSA	curto luz ultimo contato sera com imprensa	(curto, ADJ) (luz, N) (ultimo, None) (contato, N) (sera, None) (com, PREP) (imprensa, N)	4	3	(curto, adj) (luz, n) (ultimo, None) (contato, n) (sera, None) (com, prp) (imprensa, n)	4	3
61	#IDNUM FALTA DE LUZ DESDE ONTEM	falta de luz desde ontem	(falta, V) (de, PREP) (luz, N) (desde, PREP) (ontem, ADV)	5	0	(falta, v-fin) (de, prp) (luz, n) (desde, prp) (ontem, adv)	5	0
62	ESTAMOS SEM LUZ DESDE AS 9:00 DA NOITE.NUMERO DA	estamos sem luz desde 9:00 da noite	(estamos, V) (sem, PREP) (luz, N) (desde, PREP) (9:00, None) (da, NPROP) (noite, NPROP)	4	3	(estamos, v-fin) (sem, prp) (luz, n) (desde, prp) (9:00, None) (da, None) (noite, n)	5	2

#	Message	Tokenized Sentence	Mac_Morpho	Hits	Misses	Floresta	Hits	Misses
	ISTALACAO E #IDNUM ESTA NO NOME DO CLAUDINO DOS SANTOS.	numero da istalacao nome do claudino dos santos	(numero, None) (da, NPROP) (istalacao, None) (nome, N) (do, KS) (claudino, NPROP) (dos, NPROP) (santos, NPROP)	4	4	(numero, None) (da, None) (istalacao, None) (nome, n) (do, None) (claudino, None) (dos, prop) (santos, prop)	3	5
63	LUZ #IDNUM HA DOIS DIAS ESTAMOS SEM LUZ	luz ha dois dias estamos sem luz	(luz, N) (ha, N) (dois, NUM) (dias, N) (estamos, VAUX) (sem, PREP) (luz, N)	6	1	(luz, n) (ha, n) (dois, num) (dias, n) (estamos, v-fin) (sem, prp) (luz, n)	6	1
64	LUZ #IDNUM CHAVE CAIDA DEFRONTE PROPRIEDADE	luz chave caida defronte propriedade	(luz, N) (chave, ADJ) (caida, None) (defronte, V) (propriedade, N)	2	3	(luz, n) (chave, n) (caida, None) (defronte, None) (propriedade, n)	3	2
65	LUZ - CONDOMINIO 4 BLOCOS - #IDNUM#IDNUM#IDNUM #IDNUM	luz condominio 4 blocos	(luz, N) (condominio, None) (4, NUM) (blocos, N)	3	1	(luz, n) (condominio, None) (4, num) (blocos, n)	3	1
66	LUZ #IDNUM FIO D ALTA CAIDO NO CHAO	luz fio d alta caido chao	(luz, N) (fio, N) (d, N) (alta, ADJ) (caido, None) (chao, None)	2	4	(luz, n) (fio, n) (d, None) (alta, n) (caido, None) (chao, None)	3	3
67	LUZ #IDNUM FUZIL CAIDO NO TRANSFORMADOR	luz fuzil caido transformador	(luz, N) (fuzil, N) (caido, None) (transformador, ADJ)	2	2	(luz, n) (fuzil, n) (caido, None) (transformador, None)	2	2
68	LUZ POR FAVOR VE CE HOGE LUZ VEM FAZ QUATRO DIAS CEM LUZ TUDO ESTRAGANDO NAS GELADEIRA #IDNUM	luz favor ve ce hoge luz vem faz quatro dias cem luz tudo estragando geladeira	(luz, N) (favor,) (ve, None) (ce, NPROP) (hoge, None) (luz, N) (vem, V) (faz, V) (quatro, NUM) (dias, N) (cem, NUM) (luz, N) (tudo, PROSUB) (estragando, V) (geladeira, N)	10	5	(luz, n) (favor, n) (ve, None) (ce, prop) (hoge, None) (luz, n) (vem, v-fin) (faz, v-fin) (quatro, num) (dias, n) (cem, num) (luz, n) (tudo, pron-indp) (estragando, None) (geladeira, n)	10	5
69	CAIU O FUZIU DA ESTRADA DA QUERENCIA #IDNUM	caiu fuziu da estrada da querencia	(caiu, V) (fuziu, None) (da, NPROP) (estrada, N) (da, N) (querencia, None)	2	4	(caiu, v-fin) (fuziu, None) (da, None) (estrada, n) (da, None) (querencia, None)	2	4
70	5 HORA TARDE FALTOU LUZ #IDNUM	5 hora tarde faltou luz	(5, N) (hora, N) (tarde, ADV) (faltou, V) (luz, N)	4	1	(5, num) (hora, n) (tarde, n) (faltou, v-fin) (luz, n)	4	1
71	LUZ O FIO DA LUZ ESTA REBEMTADO NO MUNICIPIO DE ENCRUZILHADA 3DISTRITO TABULEIRO #IDNUM	luz fio da luz rebemtado municipio de encruzilhada 3distrito tabuleiro	(luz, N) (fio, N) (da, NPROP) (luz, NPROP) (rebemtado, None) (municipio, None) (de, PREP) (encruzilhada, N) (3distrito, None) (tabuleiro, N)	5	5	(luz, n) (fio, n) (da, None) (luz, n) (rebemtado, None) (municipio, None) (de, prp) (encruzilhada, n) (3distrito, None) (tabuleiro, n)	6	4

#	Message	Tokenized Sentence	Mac_Morpho	Hits	Misses	Floresta	Hits	Misses
72	ESTAMO CEN LUZ CAIU A FICHA DO TRANSFORMADOR NUMERO DA INSTALACAO #IDNUM	estamo cen luz caiu ficha do transformador numero da instalacao	(estamo, None) (cen, None) (luz, N) (caiu, V) (ficha, N) (do, KS) (transformador, ADJ) (numero, None) (da, NPROP) (instalacao, None)	3	7	(estamo, None) (cen, None) (luz, n) (caiu, v-fin) (ficha, n) (do, None) (transformador, None) (numero, None) (da, None) (instalacao, None)	3	7
73	LUZ #IDNUM CAIU UM POSTE COM A REDE.	luz caiu um poste com rede	(luz, N) (caiu, V) (um, ART) (poste, N) (com, PREP) (rede, N)	6	0	(luz, n) (caiu, v-fin) (um, art) (poste, n) (com, prp) (rede, n)	6	0
74	LUZ. RECRAMAR FALTA DE LUZ POSTE CAIDO .OBRIGADO POR ATENDER.	luz recramar falta de luz poste caido	(luz, N) (recramar, None) (falta, N) (de, PREP) (luz, N) (poste, N) (caido, None)	5	2	(luz, n) (recramar, None) (falta, n) (de, prp) (luz, n) (poste, n) (caido, None)	5	2
		atender	(atender, V)	1	0	(atender, v-inf)	1	0
75	TODA A CIDADE DE MARIANA PIMENTEL ESTA SEM LUZ.	toda cidade de mariana pimentel sem luz	(toda, PROADJ) (cidade, N) (de, PREP) (mariana, NPROP) (pimentel, NPROP) (sem, NPROP) (luz, NPROP)	5	2	(toda, pron-det) (cidade, n) (de, prp) (mariana, None) (pimentel, None) (sem, prp) (luz, n)	3	4
76	LUZ FIO ARREBENTADO INT #IDNUM	luz fio arrebentado int	(luz, N) (fio, N) (arrebentado, None) (int, None)	2	2	(luz, n) (fio, n) (arrebentado, None) (int, None)	2	2
77	URGENTE FIO DO POSTE DA RUA ESTA SE SOLTANDO SO MINHA CASA ESTA SEM LUZ #IDNUM	urgente fio do poste da rua soltando so minha casa sem luz	(urgente, ADJ) (fio, N) (do, KS) (poste, N) (da, N) (rua, N) (soltando, V) (so, None) (minha, PROADJ) (casa, N) (sem, PREP) (luz, N)	9	3	(urgente, adj) (fio, n) (do, None) (poste, n) (da, None) (rua, n) (soltando, None) (so, None) (minha, pron-det) (casa, n) (sem, prp) (luz, n)	8	4
78	FALTA LUZ HA MEIA HORA #IDNUM	falta luz ha meia hora	(falta, V) (luz, N) (ha, N) (meia, N) (hora, N)	3	2	(falta, v-fin) (luz, n) (ha, n) (meia, n) (hora, n)	3	2
79	LUZ FRACA SO NUMA FASE CLIENTE	luz fraca so numa fase cliente	(luz, N) (fraca, ADJ) (so, None) (numa, NPROP) (fase, N) (cliente, N)	4	2	(luz, n) (fraca, adj) (so, None) (numa, None) (fase, n) (cliente, n)	4	2
80	O BAIRRO TODO ESTA SEM LUZ	bairro todo sem luz	(bairro, N) (todo, PROADJ) (sem, PREP) (luz, N)	4	0	(bairro, n) (todo, pron-det) (sem, prp) (luz, n)	4	0
81	LUZ #IDNUM FAZEM 6 HORAS SEM LUZ	luz fazem 6 horas sem luz	(luz, N) (fazem, V) (6, NUM) (horas, N) (sem, PREP) (luz, N)	6	0	(luz, n) (fazem, v-fin) (6, num) (horas, n) (sem, prp) (luz, n)	6	0
82	LUZ #IDNUM FICHA CAIDA POSTE TAQUARI COM ESTEIO	luz ficha caida poste taquari com esteio	(luz, N) (ficha, N) (caida, None) (poste, N) (taquari, NPROP) (com, PREP) (esteio, NPROP)	6	1	(luz, n) (ficha, n) (caida, None) (poste, n) (taquari, None) (com, prp) (esteio, None)	4	3
83	LUZ #IDNUM AGORA TA SO NUMA FASE	luz agora ta so numa fase	(luz, N) (agora, KS) (ta, None) (so, None) (numa, NPROP) (fase, N)	2	4	(luz, n) (agora, adv) (ta, None) (so, None) (numa, None) (fase, n)	3	3

#	Message	Tokenized Sentence	Mac_Morpho	Hits	Misses	Floresta	Hits	Misses
84	LUZ #IDNUM CAIU CHAVE NO POSTE DA CEEE	luz caiu chave poste da ceee	(luz, N) (caiu, V) (chave, N) (poste, N) (da, NPROP) (ceee, NPROP)	5	1	(luz, n) (caiu, v-fin) (chave, n) (poste, n) (da, None) (ceee, None)	4	2
85	LUZ #IDNUM ESTAMOS SEM LUZ DESDE AS 4HS DA TARDE FAVOR RESTABELECER	luz estamos sem luz desde 4hs da tarde favor restabelecer	(luz, N) (estamos, VAUX) (sem, PREP) (luz, N) (desde, PREP) (4hs, None) (da, NPROP) (tarde, NPROP) (favor,) (restabelecer, V)	6	4	(luz, n) (estamos, v-fin) (sem, prp) (luz, n) (desde, prp) (4hs, None) (da, None) (tarde, n) (favor, n) (restabelecer, None)	7	3
86	NAO TEMOS LUZ NA RUA 2 DO DUNAS DES DAS 11 HORAS DA MANHA ESTOU LIGANDO FAZ HORAS E SO DA OCUPADO	nao temos luz rua 2 do dunas des das 11 horas da manha estou ligando faz horas so da ocupado	(nao, ADV) (temos, V) (luz, N) (rua, N) (2, N) (do, N) (dunas, None) (des, NPROP) (das, NPROP) (11, N) (horas, N) (da, NPROP) (manha, None) (estou, V) (ligando, V) (faz, V) (horas, N) (so, None) (da, NPROP) (ocupado, PCP)	10	10	(nao, None) (temos, v-fin) (luz, n) (rua, n) (2, num) (do, None) (dunas, n) (des, None) (das, None) (11, num) (horas, n) (da, None) (manha, None) (estou, v-fin) (ligando, v-ger) (faz, v-fin) (horas, n) (so, None) (da, None) (ocupado, v-pcp)	11	9
87	LUZ #IDNUM NA MINHA QUADRA TODOS JA TEM LUZ SOMENTE A MINHA CASA QUE NAO TEM E O PROBLEMA EH NO POSTE DA RUA QUE TEM UM FIO SOLTO	luz minha quadra todos tem luz somente minha casa nao tem problema eh poste da rua tem um fio solto	(luz, N) (minha, PROADJ) (quadra, N) (todos, PROADJ) (tem, V) (luz, N) (somente, PDEN) (minha, PROADJ) (casa, N) (nao, ADV) (tem, V) (problema, N) (eh, IN) (poste, N) (da, N) (rua, N) (tem, V) (um, ART) (fio, N) (solto, PCP)	17	3	(luz, n) (minha, pron-det) (quadra, n) (todos, pron-det) (tem, v-fin) (luz, n) (somente, adv) (minha, pron-det) (casa, n) (nao, None) (tem, v-fin) (problema, n) (eh, None) (poste, n) (da, None) (rua, n) (tem, v-fin) (um, art) (fio, n) (solto, None)	16	4
88	DESDE COMECO DA TARDE SEM ENERGIA LUZ #IDNUM	desde comeco da tarde sem energia luz	(desde, PREP) (comeco, None) (da, NPROP) (tarde, NPROP) (sem, NPROP) (energia, NPROP) (luz, NPROP)	1	6	(desde, prp) (comeco, None) (da, None) (tarde, n) (sem, prp) (energia, n) (luz, n)	5	2
89	LUZ ARVORE ENCIMA DOS FIOS #IDNUM	luz arvore encima dos fios	(luz, N) (arvore, None) (encima, None) (dos, NPROP) (fios, N)	2	3	(luz, n) (arvore, None) (encima, None) (dos, prop) (fios, n)	2	3
90	ESTAMOS SEM LUZ DESDE ONTEM INSTALACAO #IDNUM	estamos sem luz desde ontem instalacao	(estamos, V) (sem, PREP) (luz, N) (desde, PREP) (ontem, ADV) (instalacao, None)	5	1	(estamos, v-fin) (sem, prp) (luz, n) (desde, prp) (ontem, adv) (instalacao, None)	5	1
91	LUZ DESDE AS 17:00 HS DO DIA 19/09 #IDNUM	luz desde 17:00 hs do dia 19/09	(luz, N) (desde, PREP) (17:00, None) (hs, None) (do, NPROP) (dia, NPROP) (19/09, None)	2	5	(luz, n) (desde, prp) (17:00, None) (hs, None) (do, None) (dia, n) (19/09, None)	3	4

#	Message	Tokenized Sentence	Mac_Morpho	Hits	Misses	Floresta	Hits	Misses
92	LUZ TEM GALHOS D ARVO BATEN NOS FIOS #IDNUM	luz tem galhos d arvo baten fios	(luz, N) (tem, V) (galhos, N) (d, N) (arvo, None) (baten, None) (fios, N)	4	3	(luz, n) (tem, v-fin) (galhos, None) (d, None) (arvo, None) (baten, None) (fios, n)	3	4
93	LUZ #IDNUM AINDA ESTAMOS SEM LUZ DESDE ONTEM	luz ainda estamos sem luz desde ontem	(luz, N) (ainda, ADV) (estamos, V) (sem, PREP) (luz, N) (desde, PREP) (ontem, ADV)	7	0	(luz, n) (ainda, adv) (estamos, v-fin) (sem, prp) (luz, n) (desde, prp) (ontem, adv)	7	0
94	LUZ: TA 24H SEM LUZ JA DE MAIS ATE QUE HORA AGENTE VAI FICAR SEM LUZ ? #IDNUM	luz ta 24h sem luz de mais ate hora agente vai ficar sem luz	(luz, N) (ta, None) (24h, N|HOR) (sem, PREP) (luz, N) (de, PREP) (mais, PREP) (ate, None) (hora, N) (agente, N) (vai, VAUX) (ficar, V) (sem, PREP) (luz, N)	11	3	(luz, n) (ta, None) (24h, None) (sem, prp) (luz, n) (de, prp) (mais, pron-det) (ate, None) (hora, n) (agente, n) (vai, v-fin) (ficar, v-inf) (sem, prp) (luz, n)	11	3
95	#IDNUM SEM LUZ NA RUA JOAQUIM LOUZADA DESDE AS 18:30	sem luz rua joaquim louzada desde 18:30	(sem, PREP) (luz, N) (rua, N) (joaquim, NPROP) (louzada, None) (desde, PREP) (18:30, None)	5	2	(sem, prp) (luz, n) (rua, n) (joaquim, None) (louzada, None) (desde, prp) (18:30, None)	4	3
96	LUZ #IDNUM JA ESTAMOS SEM LUZ A 3 DIAS AGILISA AI MEU	luz estamos sem luz 3 dias agilisa ai meu	(luz, N) (estamos, VAUX) (sem, PREP) (luz, N) (3, N|AP) (dias, N) (agilisa, None) (ai, IN) (meu, PROADJ)	6	3	(luz, n) (estamos, v-fin) (sem, prp) (luz, n) (3, num) (dias, n) (agilisa, None) (ai, prop) (meu, pron-det)	7	2
97	OS A MAIS DE CINCO DIAS SEM LUZ E SEM O MINIMO RESPEITO DAS AUTORIDADES DA CEEE.	mais de cinco dias sem luz sem minimo respeito das autoridades da ceee	(mais, ADV) (de,) (cinco, NUM) (dias, N) (sem, PREP) (luz, N) (sem, PREP) (minimo, None) (respeito, N) (das, NPROP) (autoridades, NPROP) (da, NPROP) (ceee, NPROP)	8	5	(mais, adv) (de, prp) (cinco, num) (dias, n) (sem, prp) (luz, n) (sem, prp) (minimo, None) (respeito, n) (das, None) (autoridades, n) (da, None) (ceee, None)	9	4
98	LUZ POR FAVOR JA FAZ 3 DIA HOJE PODE VIM AGORA ARUMAR -SIM OU NAO- #IDNUM	luz favor faz 3 dia hoje pode vim agora arumar sim ou nao	(luz, N) (favor,) (faz, V) (3, NUM) (dia, N) (hoje, ADV) (pode, VAUX) (vim, V) (agora, ADV) (arumar, None) (sim, ADV) (ou, ADV) (nao, ADV)	11	2	(luz, n) (favor, n) (faz, v-fin) (3, num) (dia, n) (hoje, adv) (pode, v-fin) (vim, v-fin) (agora, adv) (arumar, None) (sim, adv) (ou, conj-c) (nao, None)	11	2
99	LUZ #IDNUM JA FAZ MAIS DE 3 HORAS QUE EU ESTOU AGUARDANDO	luz faz mais de 3 horas eu estou aguardando	(luz, N) (faz, V) (mais, ADV) (de, PREP) (3, NUM) (horas, N) (eu, PROPESS) (estou, VAUX) (aguardando, V)	9	0	(luz, n) (faz, v-fin) (mais, adv) (de, prp) (3, num) (horas, n) (eu, pron-pers) (estou, v-fin) (aguardando, v-ger)	9	0

#	Message	Tokenized Sentence	Mac_Morpho	Hits	Misses	Floresta	Hits	Misses
100	LUZ #IDNUM ESTA FALTANDO LUZ A 5 DIAS O PROBLEMA E NOS FUZIVEIS AS CAMI ONETES PASAM DIRETO	luz faltando luz 5 dias problema fuziveis cami onetes pasam direto	(luz, N) (faltando, V) (luz, N) (5, N|AP) (dias, N) (problema, N) (fuziveis, None) (cami, None) (onetes, None) (pasam, None) (direto, ADJ)	6	5	(luz, n) (faltando, v-ger) (luz, n) (5, num) (dias, n) (problema, n) (fuziveis, None) (cami, None) (onetes, None) (pasam, None) (direto, adj)	7	4
			Total:	614	280	**Total:**	612	282
			Average:	0,6868		**Average:**	0,6845	

APPENDIX C – NOTIFICATION TYPES' VALIDATION QUESTIONNAIRE

The following questionnaire was presented to a group of domain expert judges in order to validate the categories of events and notification types defined for the case study.

Avaliação de Níveis de Severidade - Tipos de Notificação

O objetivo deste questionário é determinar o grau de severidade dos tipos de notificação dos clientes à companhia.

Para isto, foi definido um recorte com 13 tipos de informação, organizados em três categorias. A categoria 'Instalação' compreende notificações referentes apenas a instalação do cliente. A categoria 'Rede' representa notificações que podem afetar a rede elétrica, e por consequência, um número maior de clientes. Por fim, a categoria 'Ambiente' indica notificações sobre o local e condições climáticas que possam afetar os clientes e a manutenção dos serviços.
Para cada resposta, deve-se informar o grau de severidade (1 a 5, sendo 5 o maior nível de severidade) para cada tipo de notificação.

* Required

1. **Falta de Luz (Instalação)** *
   Notificação de falta de energia.

2. **Meia Fase (Instalação)** *
   Notificação de queda de tensão na instalação.

3. **Oscilação (Instalação)** *
   Notificação de oscilação da energia elétrica na instalação do cliente.

4. **Fio e Cabos Partidos (Rede)** *
   Rompimento de fio elétrico, prejudicando o fornecimento de energia aos clientes.

5. **Transformador Caído (Rede)** *
   Avaria no Transformador que prejudique o bom funcionamento da rede.

6. **Chave Fusível Desligada** *
   Problema encontrado na Chave Fusivel que atende o cliente notificador.

7. **Curto Circuito (Rede)** *
   Dano causado na rede elétrica da companhia.

8. **Queda de Poste (Rede)** *
   Notificação de poste caído.

9. **Fogo na Rede (Rede)** *
   Notificação de incêndio na rede elétrica.

10. **Queda de Árvore (Ambiente)** *
    Notificação de queda de árvore nos fios ou postes da companhia.

11. **Raio (Ambiente)** *
    Notificação de raio(s) que pode(m), além de atingir a rede, atingir áreas maiores e condicionar o atendimento na área.

12. **Vento (Ambiente)** *
    Notificação de condição climática desfavorável, podendo prejudicar a rede elétrica e dificultar o atendimento na região.

13. **Chuva (Ambiente)** *
    Notificação de condição climática desfavorável, podendo prejudicar a rede elétrica e dificultar o atendimento na região.

14. **Sugestões - Tipos de Notificação**
    Alguma das categorias ou tipos de notificação definidos deveriam ser modificadas ou removidos? Por quê?

15. **Sugestões - Definições**
    Existe alguma categoria ou tipo de notificação relevante que deveria ser considerada?

APPENDIX D – SMS MESSAGES' TAGGING QUESTIONNAIRE

This questionnaire comprises a set of 100 SMSs from BraCorpSMS with the purpose of creating a gold standard to evaluate the prototype's temporal and event taggers. In each question, the judges should check whether the messages contain information corresponding to the model of categories of events as well as temporal information.

Avaliação das Mensagens SMS conforme as Notificações

O objetivo deste questionário é verificar se as notificações discutidas no questionário anterior se encontram nestas mensagens.
Cada questão corresponde a uma mensagem SMS. Assinale a(s) opção(ões) que considerar presente em cada uma das frases. As notificações não são exclusivas, ou seja, uma mensagem pode conter vários tipos de notificação. Caso nenhuma das notificações seja adequada para a mensagem, ou caso queira justificar suas escolhas, selecione a opção "Others" e então deixe um comentário.

Com o cruzamento das marcações feitas neste questionário, será definida a marcação "correta" para cada mensagem, a qual será comparada com a marcação do protótipo já construído. Dessa forma, é extremamente importante que as questões sejam preenchidas com muito cuidado.

É permitido editar as respostas após o envio do formulário.

Considere que:
Informação Temporal corresponde à qualquer expressão indicando tempo, como 'desde ontem', 'há 12h', 'meio dia', 'fevereiro', 'Natal', etc.
'Instalação' compreende notificações referentes apenas à instalação do cliente.
'Rede' representa notificações que podem afetar a rede elétrica, e por consequência, um número maior de clientes.
'Ambiente' indica notificações sobre o local e condições climáticas que possam afetar os clientes e a manutenção dos serviços.
#IDNUM corresponde ao número da instalação do cliente, devidamente anonimizado

Muito obrigado pela colaboração!

* Required

1. **Mensagem #01** *
   LUZ #IDNUM CAIU A FICHA
   Check all that apply.

   - [ ] Informação Temporal
   - [ ] Instalação - Falta de Luz
   - [ ] Instalação - Meia Fase
   - [ ] Instalação - Oscilação
   - [ ] Rede - Fio e Cabos Partidos
   - [ ] Rede - Transformador Caído
   - [ ] Rede - Chave Fusível Desligada
   - [ ] Rede - Curto Circuito
   - [ ] Rede - Queda de Poste
   - [ ] Rede - Fogo na Rede
   - [ ] Ambiente - Queda de Árvore
   - [ ] Ambiente - Raio
   - [ ] Ambiente - Vento
   - [ ] Ambiente - Chuva
   - [ ] Other: .....

APPENDIX E – GOLD STANDARD EVALUATION

Here, we present the gold standard, where each cell represents a different message and each number represents an identified notification type. Furthermore, we show the prototype output, in the same format, as it was used for the Evaluation of the temporal and event taggers.

ID	Gold Std.	Prototype	ID	Gold Std.	Prototype	ID	Gold Std.	Prototype
420257	2	2	462510	2,5,8	0	490495	2	2
420555	2,12	2	462672	2,11	2,11	494530	2,5,8	2,8
421280	2	2	463267	2,5,9	2	494626	2,6	9
423346	1,2	1,2	464362	2,11	2	494979	2,10,11	2,1
424385	2,5	2	464593	2,6	2	522232	2,13	2
426883	1,2	1,2	466855	2,5	2,5	511110	2,13	1,2,5
427452	1,2	1,2	468719	1,2,3	2	503472	2,5	2
427566	1,2	1	469315	4	2	504186	2,6	2
428849	2,11	2	470688	2,5	2	504972	9	0
429275	1,2	1,2	472481	11	2	505202	2,9,11	2,9
430780	2,4	1	473499	2,10,11	2	505325	1,2,5	1,2
433808	2,7	2	473666	2,1	2,1	505977	1,2	1,2
434725	2,7	2	473966	9	0	506084	2,5,9,11,13	2,9,11
435562	3	2	475171	2,1	2,1	512324	1,2,5	1,2
440916	2,3	2	476512	1,2	1,2	512837	1,2	1,2
443656	2,5,11	2,11	477053	10	10	514331	1,2	2
444076	1,2	1,2	477675	1,2	2	515173	2,3	2
444868	2,7	2	478608	2,5	2	515498	1,2,5,8	1,2
445618	13,14	0	478821	2,11	2	527112	2,7	2,7
446035	2,5	2,5	478896	2,5	5	527184	1,2,5	1,2,5
447773	1,2	1,2	479408	2,5	2	537390	1,2,5,8	1,2
448656	3,4	2	479526	2	2	543853	1,2	1
451820	2,5	2	479720	2,9	2	549712	0	0
452284	14	2	480612	2,5,9	2	550378	1,2	2
452468	3	2	481032	1,2	1,2	550513	2,5	2,8
453346	2	2	481989	2,9	2	550751	1,2	1,2
454684	2,1	2,1	482088	2,5	2	545888	1,2	2
454775	2,5	2	483161	2,5	2,5	545557	2,3	2
455698	2,9	2	483628	2,5	2	539984	1,2	1,2
456459	2,1	2	484190	2,5,11	2,11	531029	1,2	1,2
458582	2,7	2	486899	9	9	527756	2,3,5,8	2,8
458989	2,5	2	488157	1,2,5	1,2	523648	1,2	1,2
461948	2,5,11	11	489925	2,5	2			
462271	2,5	2,5	490145	2,3	0			