Towards Definition Extraction Using Conditional Random Fields

Luis Espinosa Anke
Universitat Pompeu Fabra
luis.espinosa83@gmail.com

Abstract

Definition Extraction (DE) and terminology are contributing to help structuring the overwhelming amount of information available. This article presents KESSI (Knowledge Extraction System for Scientific Interviews), a multilingual domain-independent machine-learning approach to the extraction of definitional knowledge, specifically oriented to scientific interviews. The DE task was approached as both a classification and a sequential labelling task. In the latter, figures of Precision, Recall and F-Measure were similar to human annotation, and suggest that combining structural, statistical and linguistic features with Conditional Random Fields can contribute significantly to the development of DE systems.

1 Introduction

We present and discuss the process of building and evaluating a DE system for educational purposes. Aimed at exploiting the genre of scientific interviews, and envisaged as a time-saving tool for semi-automatically creating listening comprehension exercises, we present a Knowledge Extraction System for Scientific Interviews (KESSI). It is based on the theoretical and methodological foundations of DE, the task to automatically identify definitional sentences within texts (Navigli and Velardi, 2010).

KESSI is a DE system that relies solely on machine-learning techniques, which has the advantage of overcoming the domain-specificity and language dependence of rule-based methods (Del Gaudio et al., 2013). In order to train and test our model, the SMPoT (*Science Magazine Podcast Transcripts*) corpus was compiled and annotated with linguistic, terminologic and definitional information.

Two main contributions emerge from the work here presented. Firstly, it provides an analysis and discussion of the genre of scientific interviews, and examines its potential for NLP applications. We hypothesize that these interviews constitute a valuable source of information, as many scientific disciplines are covered, but dealt with in a standard register rather than the highly formal and structured register of technical manuals or scientific papers or books. Scientific interviews also present the audience with turntaking, courtesy and pragmatic elements that can prove useful for linguistic research as well as the development of Natural Language Processing tools. Secondly, promising results that border or go beyond 90% in Precision and Recall demonstrate that using CRF for DE is a viable option. These results also seem to suggest that combining linguistic information (surface forms, Part-of-Speech and syntactic functions), statistical information (word counts or tf-idf) and structural information (position of the token within the document, or whether it is the interviewer or the interviewee who speaks) can contribute to the design of DE systems.

2 Related Work

It can be argued that in general, most approaches to automatic DE rely on rule-based methods. These have ranged from verb-matching (Rebeyrolle and Tanguy, 2000; Saggion and Gaizauskas, 2004; Sarmento et al., 2006; Storrer and Wellinghoff, 2006) to punctuation (Muresan and Klavans, 2002; Malaisé et al., 2004; Sánchez and Márquez, 2005; Przepiórkowski et al., 2007; Monachesi and Westerhout, 2008) or layout features (Westerhout, 2009). It seems reasonable to argue that there are three main problems when approaching DE as a pattern-matching task (Del Gaudio et al., 2013): Firstly, it is necessary to start almost from scratch, as it is necessary to look for specific patterns which appear

Proceedings of the Student Research Workshop associated with RANLP 2013, pages 63–70,
Hissar, Bulgaria, 9-11 September 2013.

Feature	Description
Pairs word-lemma	In a two-word window, we look at combinations surface form + lemma. In our example, this would be [it + ,], [it + lasts], [it + last], [last + essentially], and so on.
Pairs lemma + POS	In a two-word window, we would retrieve features like [it + V_PRES_SG3], [V_PRES_SG3 + essentially] or [essentially + ADV].
Who speaks	We focus on who mentions the current token. In our example, the interviewee.
Tf-Idf + surface form + lemma	In a two-word window, we would retrieve features like [3.32 + lasts + essentially] or [3.64 + essentially + forever]. Note that it it is possible to retrieve features from instances that are after the current token.

Table 1: Some of the features used for training the CRF model.

repeatedly in definitions. Secondly, these rules are language-dependent. Thirdly, they are also domain-dependent, making it difficult to extend them beyond the domain of application to which they were initially intended.

In order to overcome these problems, machine-learning techniques can be incorporated to the process. The most widely used algorithms have been Naïve Bayes, Maximum Entropy or Support Vector Machines, in the case of Fahmi and Bouma (2006), Naïve Bayes and Maximum Entropy (Rodríguez, 2004), genetic algorithms (Borg, 2009) or balanced random forests, in Degórski et al. (2008a; 2008b) and Westerhout (2010). Concerning unsupervised approaches, Zhang (2009) used a bootstrapping algorithm for the extraction of definitions in Chinese.

3 The SMPoT Corpus: Compilation and Annotation

We design a corpus following the criteria elicited by McEnery and Wilson (2001). The corpus consists of 50 fully annotated interview transcripts. Table 2 summarizes the size of the corpus in terms of words, sentences, terms and definitions.

Unit type	Count
Words	389293
Sentences	15315
Terms	26194
Definitions	570

Table 2: Raw counts for the SMPoT corpus

3.1 Preprocessing

After manually downloading and converting the pdf files from the *Science Magazine Website*[1], these were parsed using the dependency parser *Machinese Syntax* (Tapanainen and Järvinen, 1997). In this way, linguistic information such as lemma, Part-of-Speech, syntactic functions or a word's position in a dependency tree is provided.

Once the documents were collected, converted, pre-processed and automatically parsed, the next step was to semi-automatically annotate the terminology. For this, we benefited from an API for Python of the Yahoo! Term Extractor (also known as Yahoo! Content Analysis [2]). Terms were identified, and `<Term></Term>` tags were inserted to the xml document. Since terms can span multiple words, the `<Term></Term>` tags were introduced as parent nodes of the `<token>` tags. When queried, the Term Extractor API yields a list of terms, but its results depend on the size of the input text. This means that each document of the corpus had first to be split in sentences, and then each sentence was queried in order to preserve a high recall.

3.2 Annotating Definitions

This annotation schema builds up on previous work by Sierra et al. (2006) and Westerhout and Monachesi (2007). It is argued that in a textual genre like scientific interviews, where a certain degree of specificity and technical jargon is present,

[1] http://www.sciencemag.org/site/multimedia/podcast/

[2] http://developer.yahoo.com/contentanalysis

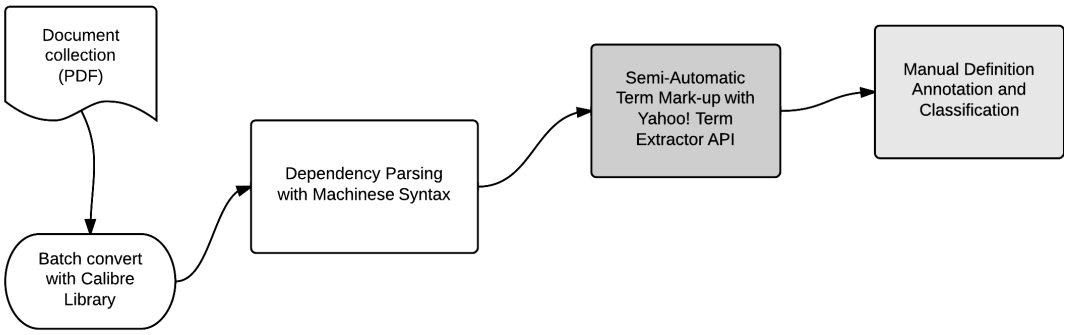

Figure 1: Summary of the steps involved in the compilation and annotation of the corpus.

a classification that looks at the patterns of the definitions alone, or at their information alone, might prove insufficient to capture the complexity of the way information is presented. Table 3 shows the 5 most frequent types of this two-dimensional classification, as well as their count and an example of each.

So far, the annotation process (summarized in Figure 1) has been examined, which consisted in automatic linguistic markup, semi-automatic terminology identification, and manual definition labelling and classification.

4 The Development of KESSI

Once the dataset is compiled and enriched, and can be used for training and testing purposes, we approach the DE task as (1) a binary classification task, where each sentence is labeled as *has_def* or *no_def*, and (2) a sequential labeling task, where each token is tagged according to whether it is *Inside*, *Outside*or at the *Beginning* of a definitional clause.

4.1 Binary Classification

Using the Weka workbench (Witten and Frank, 2005), we train a set of machine-learning algorithms in order to classify unseen sentences as containing or not containing a definition. However, a previous step seems necessary in order to handle properly the imbalanced dataset issue. According to Del Gaudio et al. (2013), few works have specifically addressed this issue through some kind of sampling. We take an approach similar to Degórski et al. (2008b), where a number of subsampled training datasets are used to increase the ratio of positive instances, specifically 1:1, 2:1 and 5:1. Moreover, simple linguistically motivated features were used. We extracted the 500 most frequent ngrams (`n = 1, 2, 3`), and used the linguistic information provided by the parser. This resulted in 1-3grams for surface forms, Part-Of-Speech and syntactic functions. In addition, we also added pattern-based features, like the presence or absence of the sequence "which is" or having a term followed by the verb "to be". Finally, the algorithms selected were Naïve Bayes, Decision Trees, SVM, Logistic Regression and Random Forests.

4.2 Sequential Labelling

Building up on the premise that both linguistic and structural features can be exploited for automatic DE, we propose a method to label each token in a sequence with `B_DefClause`, `I_DefClause` or `O_DefClause` tags (which correspond to whether a token is a the beginning, inside or outside a definition). For each sentence, each token has been manually annotated with these tags. Whenever a sequence of words that form a definition is found (what we refer as Definitional Clause), the tokens that are part of it are additionally labelled as Beginning, Inside or Outside for three more categories: Term, Definitional Verb and Definition. See Figure 2 for an illustrative example of this two-layered annotation schema.

4.2.1 Conditional Random Fields

Conditional Random Fields (Lafferty and McCallum, 2001) have been used extensively in NLP, e.g.

Type of Definition	Frequency	Example
Pattern type = is def Information type = intensional	135	Clicker's an electronic response device that's keyed to the instructors computer, so the instructor is getting an answer and can grade it.
Pattern type = verb def Information type = functional	111	Mice develop regulatory T- cells against non-inherited maternal alloantigens as a result of fetal exposure.
Pattern type = verb def Information type = extensional	52	Nano-ear is made from a microscopic particle of gold that is trapped by a laser beam.
Pattern type = is def Information type = functional	44	Iridium is not very common on Earth, but it is very common in asteroids.
Pattern type = punct def Information type = synonymic	32	(...) female determinant gene, S-ribonuclease gene.

Table 3: Most common types of definitions according to a Pattern/Information-based classification

Chinese Word Segmentation (Sun et al., 2013), Named Entity Recognition (Fersini and Messina, 2013), Sentiment Analysis (Jakob and Gurevych, 2010) or TimeML event recognition (Llorens et al., 2010). They are undirected graphical models where the dependencies among input variables x do no need to be explicitly represented. This allows to use richer and more global features of the input data, e.g. features like Part-of-Speech or ngram features of surrounding words.

4.2.2 Feature Selection

The definition of features is crucial for the architecture of the system (Llorens et al., 2010). We hypothesize that combining linguistic, statistic and structural information can contribute to the improvement of DE systems. For each token, these are the features extracted:

- **Term Frequency**: Raw count for the current token within the document.
- **Tf-idf**: Relative frequency score, which takes into account not only the token count within the current document, but its spread across the collection.
- **Token index**: The position of the token in the document.
- **Is_term**: Whether the token is a term or not.
- **Surface form**: The surface form of the token.
- **Lemma**: The token's lemma. In the case of extremely highly collocated multiword units, Machinese Syntax groups them together in one token. They are left as-is, regardless of potential capitalization.
- **Part-of-Speech**: Part-of-Speech of the token, including subtypes and number.
- **Syntactic Function**: Following a dependency grammar.
- **Who speaks**: Whether it is the interviewer, the interviewee, or a dangling token, in which case it is tagged as narrator.
- **BIO_term**: Regardless of the is_term label, we also investigate the token's position within a term BIO tagging scheme.
- **BIO_DefVerb**: Labels the connecting verb between a term and a definition.
- **BIO_Definition**: Labels the chunk that constitutes actual the definition.

Since CRF allow the encoding of long-distance relations, these features are combined in order to capture relevant combinations of features occurring before and after the current token (see Table 4).

5 Evaluation

The performance of KESSI was evaluated from two different perspectives. The reason for this being that it was necessary to account for the two approaches (binary classification and sequential labelling), on one hand, and the ultimate purpose of the system, on the other. Firstly, figures of Precision, Recall and F-Measure are provided and discussed for the classification approach, consider-

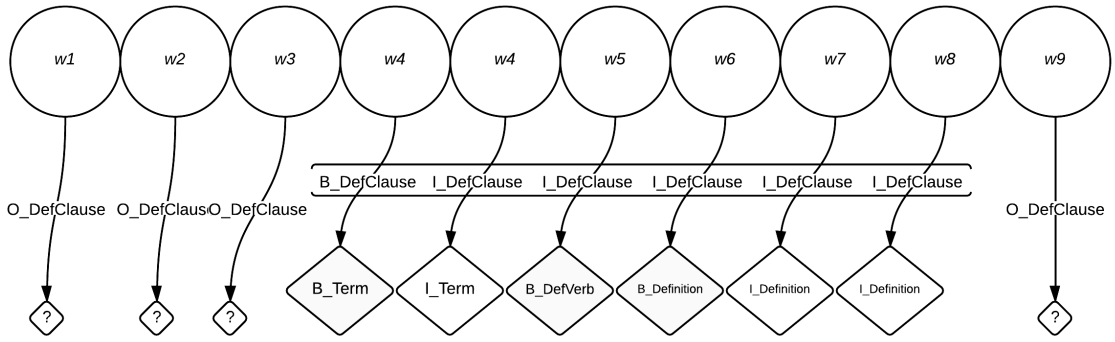

Figure 2: Visualization of the tagging schema.

Feature	Description
Pairs word-lemma	In a two-word window, we look at combinations surface form + lemma. In our example, this would be [it + ,], [it + lasts], [it + last], [last + essentially], and so on.
Pairs lemma + POS	In a two-word window, we would retrieve features like [it + V_PRES_SG3], [V_PRES_SG3 + essentially] or [essentially + ADV].
Who speaks	We focus on who mentions the current token. In our example, the interviewee.
Tf-Idf + surface form + lemma	In a two-word window, we would retrieve features like [3.32 + lasts + essentially] or [3.64 + essentially + forever]. Note that it it is possible to retrieve features from instances that are after the current token.

Table 4: Some of the features used for training the CRF model.

ing different resampling setups as well as different algorithms. Finally, Precision, Recall a nd F-Measure are reported on a high-granularity basis, in a hard evaluation, where only exact matching of a token was considered a true positive.

5.1 Classification Approach

Firstly, we examine results obtained with a simple ngram feature selection, where the 500 most frequent surface form uni, bi and trigrams are used as features for each sentence vector. Subsampling was carried out because we were more interested in correctly extracting positive instances, i.e. increasing Recall in is_def sentences. The highest F scores for positive instances were obtained under the following configurations:

1. Naïve Bayes - Original Dataset 10-fold Cross validation
2. Decision Trees - Subsample 2:1 - Test on original dataset

In setup (I), 207 positive instances out of 570 were correctly extracted, which yields a Recall of .36 for positive instances. However, by subsampling the training set to a 1:1 ratio (i.e. randomly removing negative instances until the remaining set contains the same number of positive and negative instances), it is possible to increase the desired results. As this approach cannot be tested by cross-validation, a supplied test set from the original dataset is used for testing. This test set did not overlap with the training set.

In (II), Recall increases to up to .6, as the system correctly extracts 66 out of 110 positive instances. Precision, however, remains low (P = .16). By incorporating more features where POS and syntactic functions are combined, we increase

	Original	S-1000	S-10000
All-S	P=0.97; R=0.89; F=0.93	P=0.03; R=0.98; F=0.07	P=0.08; R=0.48; F =0.15
1-S	P=0.97; R=0.90; F=0.93	P=0.03; R=0.99; F=0.06	P=0.47; R=0.95; F=0.63

Table 5: Results for the token-wise evaluation of KESSI

Recall in positive instances. For example, SVM trained with a 1:1 subsample training set shows an increase of up to .78. The effect this has on Precision is that it lowers it to .11. Finally, let us highlight the setup that obtained the highest recall for positive instances: Naïve Bayes algorithm trained with a subsampled 1:1 training set. Recall reaches .89, with the consequent drop in precision to .07.

We can conclude that combining surface form, Part-of-Speech and syntactic functions ngrams as features in a subsampled training set of 1:1 serves as the highest performing model. We consider a good model the one that correctly classifies the highest number of positive instances (i.e. those sentences that contain a definition), with the minimum loss with respect to negative instances.

5.2 CRF Evaluation

We propose a token-wise evaluation where each word is matched against the gold standard. If its `BIO_DefClause` tag does not match, it is considered incorrect. This has the advantage of knowing beforehand how many tokens we have, which is crucial for being able to compute Precision, Recall and F-Measure. It could be argued, however, that such approach is too restrictive, as it will consider as incorrect a `B_DefClause` token even if it is compared with an `I_DefClause` token, and this might not be always as accurate. In Table 5, the performance of KESSI is shown for three different sampling setups: Original train-set (Original), subsample of negative sentences down to 1000 (S-1000), and subsample of negative sentences down to 10000 (S-10000). For testing, a cut-off of the same size as in the Classification approach is used. Our test sets contain 20% of the overall positive instances, which in this case are either `B_DefClause` or `I_DefClause` tokens. This amounts to 111 definitions. Our test set consisted in, first, a dataset where all sentences are split according to their original format (All-S), and second, a dataset where all the instances are put together with no sentence boundary among them (1-S).

These results reveal that a radical resampling (leaving only 1000 negative instances), when using Conditional Random Fields, does not have a dramatic effect in performance. While Recall increases almost a 10% (from 0.89 to 0.98), Precision suffers from a strong decrease, in this case 94% (from 0.97 to 0.03). With scores nearing or above 90% in Precision, Recall and F-Measure, it seems safe to assume that using linguistic, statistic and structural features combined with CRF improve dramatically a DE system. In comparison with previous work in this field, where most datasets consisted in more structured text than interview transcripts, it also seems reasonable to claim that this method is better suited for more unstructured language.

6 Conclusions

Different stages involved in the design and development of a DE system have been presented. Once the criteria for the taxonomy were clear, an annotation task was carried out on 50 documents from The Science Magazine Podcast, where linguistic information, terminology and definitions were identified and classified. Then, the DE task was approached both as a classification problem and as a sequential labelling problem, and Precision, Recall and F-Measure results indicate that combining linguistic, structural and statistic features with Conditional Random Fields can lead to high performance. We propose the following directions for future work: Firstly, expanding the size and the dataset and incorporating additional features to the definition classification. Secondly, trying additional resampling techniques like the SMOTE algorithm in order to oversample the minority class. This algorithm has been applied successfully in this field (Del Gaudio et al., 2013). Thirdly, ensuring a more reliable annotation by incorporating additional annotators and computing some kind of agreement metric would seem advisable as in some cases a false positive might be due to the fact that the annotator missed a good definition. And finally, providing sentence-wise evaluation scores for the CRF approach, so that the two methods showcased could be evenly compared.

References

Claudia Borg. 2009. *Automatic Definition Extraction Using Evolutionary Algorithms*. MA Dissertation. University of Malta.

Lukasz Degórski, Lukasz Kobyliński and Adam Przepiórkowski. 2008. Definition extraction: improving balanced random forest. Proceedings of the International Multiconference on Computer Science and Information Technology, pp. 353-357

Lukasz Degórski, Michał Marcińczuk and Adam Przepiórkowski. 2008. Definition extraction using a sequential combination of baseline grammars and machine learning classifiers. Proceedings of the Sixth International Conference on Language Resources and Evaluation, LREC 2008.

Rosa Del Gaudio, Gustavo Batista and António Branco. 2013. Coping with highly imbalanced datasets: A case study with definition extraction in a multilingual setting. *Natural Language Engineering, pp. 1–33*

Luis Espinosa. *Forthcoming*. Classifying Different Definitional Styles for Different Users. Proceedings of CILC 2013, Alicante, 14-16 March 2013. Procedia Social and Behavioral Science. Elsevier ISSN: 1877-0428.

Ismail Fahmi and Gosse Bouma. 2006. Learning to identify definitions using syntactic features. Proceedings of the EACL 2006 workshop on Learning Structured Information in Natural Language Applications, pp. 64–71)

Elisabetta Fersini and Enza Messina. 2013. Named Entities in Judicial Transcriptions: Extended Conditional Random Fields. *Proceedings of the 14th international conference on Computational Linguistics and Intelligent Text Processing - Volume Part I* (CICLing'13), Alexander Gelbukh (Ed.), Vol. Part I. Springer-Verlag, Berlin, Heidelberg, (pp. 317–328).

Niklas Jakob and Iryna Gurevych. 2010. Extracting opinion targets in a single-and cross-domain setting with conditional random fields. In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (pp. 1035–1045). Association for Computational Linguistics.

John Lafferty and Andrew McCallum. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proceedings of the Eighteenth International Conference on Machine Learning (pp. 282-289).

Héctor Llorens, Elena Saquete and Borja Navarro-Colorado. 2010. TimeML events recognition and classification: learning CRF models with semantic roles. In Proceedings of the 23rd International Conference on Computational Linguistics (pp. 725–733). Association for Computational Linguistics.

Tony McEnery and Andrew Wilson. 2001. *Corpus Linguistics: An Introduction*. Edinburgh University Press.

Véronique Malaisé, Pierre Zweigenbaum and Bruno Bachimont. 2004. Detecting semantic relations between terms in definitions. In COLING (pp. 55–62).

Paola Monachesi and Eline Westerhout. 2008. What can NLP techniques do for eLearning?. In Proceedings of the International Conference on Informatics and Systems (INFOS08). Cairo. Egypt.

Smaranda Muresan and Judith Klavans. 2002. A method for automatically building and evaluating dictionary resources. In Proceedings of the Language Resources and Evaluation Conference (Vol. 1, No. 8, p. 30).

Roberto Navigli and Paola Velardi. 2010. Learning word-class lattices for definition and hypernym extraction. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (pp. 1318–1327). Association for Computational Linguistics.

Adam Przepiórkowski, Lukasz Degórski, Miroslav Spousta, Kiril Simov, Petya Oseneva, Lothar Lemnitzer, Vladislav Kuboň and Beata Wójtowicz. 2007. Towards the automatic extraction of definitions in Slavic. In Proceedings of the Workshop on Balto-Slavonic Natural Language Processing: Information Extraction and Enabling Technologies (pp. 43–50). Association for Computational Linguistics..

Josette Rebeyrolle and Ludovic Tanguy. 2000. *Repérage automatique de structures linguistiques en corpus: le cas des énoncés définitoires. Cahiers de grammaire*, 25:153–174.

Carlos Rodrguez. 2004. *Metalinguistic information extraction from specialized texts to enrich computational lexicons*. PhD Dissertation. Barceona: Universitat Pompeu Fabra.

Horacio Saggion and Robert Gaizauskas. 2004. Mining on-line sources for definition knowledge. In Proceedings of the 17th International FLAIRS Conference (pp. 45-52).

Alexy J. Sánchez and Melva J. Márquez R. 2005. Hacia un sistema de extracción de definiciones en textos jurídicos. Actas de la 1er Jornada Venezolana de Investigación en Lingüística e Informática. Venezuela..

Luiís Sarmento, Belinda Maia, Diana Santos, Ana Pinto and Luís Cabral. 2006. Corpógrafo V3 From Terminological Aid to Semi-automatic Knowledge Engineering. In Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006) (pp. 1502–1505).

Gerardo Sierra, Rodrigo Alarcón, César Aguilar and Alberto Barrón. 2006. Towards the building of a corpus of definitional contexts. In Proceedings of

the 12th EURALEX International Congress, Torino, Italy (pp. 229–40).

Angelika Storrer and Sandra Wellinghoff. 2006. Automated detection and annotation of term definitions in German text corpora. In Proceedings of LREC (Vol. 2006).

Xu Sun, Yaozhong Zhang, Takuya Matsuzaki, Yoshimasa Tsuruoka and Jun'ichi Tsujii. 2013. Probabilistic Chinese word segmentation with non-local information and stochastic training. *Information Processing & Management, 49(3):pp. 626–636*.

Pasi Tapanainen and Timo Järvinen. 1997. A non-projective dependency parser. In Proceedings of the fifth conference on Applied natural language processing (pp. 64–71). Association for Computational Linguistics..

Eline Westerhout. 2009. Extraction of definitions using grammar-enhanced machine learning. In Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop (pp. 88–96).

Eline Westerhout. 2010. *Definition extraction for glossary creation: a study on extracting definitions for semi-automatic glossary creation in Dutch*. PhD Dissertation. Utrecht University.

Eline Westerhout and Paola Mnachesi. 2007. Extraction of Dutch definitory contexts for elearning purposes. In Proceedings of the Computational Linguistics in the Netherlands (CLIN 2007), Nijmegen, Netherlands, (pp. 219–34).

Ian H. Witten and Eibe Frank. 2005. *ata Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

Chunxia Zhang and Peng Jiang. 2009. Automatic extraction of definitions. Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on (pp. 364–368). IEEE..