Event Detection, version 1

Deliverable D4.2.1

Version FINAL

Authors: Rodrigo Agerri[1], Itziar Aldabe[1], Zuhaitz Beloki[1], Egoitz Laparra[1], Maddalen Lopez de Lacalle[1], German Rigau[1], Aitor Soroa[1], Marieke van Erp[2], Piek Vossen[2], Christian Girardi[3] and Sara Tonelli[3]

Affiliation: (1) EHU, (2) VUA, (3) FBK

Building structured event indexes of large volumes of financial and economic data for decision making

ICT 316404

Grant Agreement No.	316404
Project Acronym	NEWSREADER
Project Full Title	Building structured event indexes of large volumes of financial and economic data for decision making.
Funding Scheme	FP7-ICT-2011-8
Project Website	http://www.newsreader-project.eu/
Project Coordinator	Prof. dr. Piek T.J.M. Vossen VU University Amsterdam Tel. + 31 (0) 20 5986466 Fax. + 31 (0) 20 5986500 Email: piek.vossen@vu.nl
Document Number	Deliverable D4.2.1
Status & Version	FINAL
Contractual Date of Delivery	September 2013
Actual Date of Delivery	November 10, 2013
Type	Report
Security (distribution level)	Public
Number of Pages	77
WP Contributing to the Deliverable	WP4
WP Responsible	EHU
EC Project Officer	Susan Fraser

Authors: Rodrigo Agerri[1], Itziar Aldabe[1], Zuhaitz Beloki[1], Egoitz Laparra[1], Maddalen Lopez de Lacalle[1], German Rigau[1], Aitor Soroa[1], Marieke van Erp[2], Piek Vossen[2], Christian Girardi[3] and Sara Tonelli[3]

Keywords: Event detection, EN pipelines, NL pipeline, ES pipeline, IT pipeline, Scaling of text processing

Abstract: This deliverable describes the first prototype for event detection. It is focused on English language and it uses an open architecture. It works with generic NLP modules that perform different tasks for event detection. Each task is executed by one module, which allows custom pipelines for text processing. The design of the Dutch, Italian and Spanish pipelines are also presented. The design framework for testing the scaling capabilities of our NLP processing pipelines are also described in this document.

Table of Revisions

Version	Date	Description and reason	By	Affected sections
0.1	19 July 2013	Structure of the deliverable set	Rodrigo Agerri, Itziar Aldabe, Egoitz Laparra, Maddalen Lopez de Lacalle, German Rigau, Aitor Soroa	All
0.2	16 September 2013	Added information of introduction, event detection and English NLP processing	Itziar Aldabe	1, 2, 3
0.3	20 September 2013	Added information of event classification and Dutch pipeline. Revision of event detection	Piek Vossen	2, 3, 4
0.4	20 September 2013	Revision of introduction	German Rigau	2
0.5	23 September 2013	TextPro based pipeline added	Christian Girardi	3
0.6	24 September 2013	Added scalability issues	Zuhaitz Beloki, Aitor Soroa	7
0.7	24 September 2013	Added descriptions of factuality and discourse. Revision of NLP modules	Marieke van Erp	3
0.8	25 September 2013	Added IXA pipeline; Revision of NLP modules; Spanish pipeline added	Itziar Aldabe	3
0.9	30 September 2013	Revision of NLP modules; Italian pipeline added	Christian Girardi, Sara Tonelli	3, 5
1.0	30 September 2013	Revision of introduction, event detection and conclusions	German Rigau	1, 2, 8
1.1	10 November 2013	Full revision and update	Marieke van Erp, Itziar Aldabe, German Rigau	All

Executive Summary

This deliverable describes the first cycle of event detection, developed within the European FP7-ICT-316404 "Building structured event indexes of large volumes of financial and economic data for decision making (NewsReader)" project. The prototype and results presented are part of the activities performed in tasks T4.1 Language Resources and Processors, T4.2 Event Detection, T4.3 Authority and factuality computation and T4.5 Scaling of text processing of Work Package WP4 (Event Detection).

The first prototype on the event detection mainly focuses on English language. It works with generic NLP modules that perform tokenization, POS-tagging, parsing, time recognition, named entity recognition, word sense disambiguation, named entity disambiguation, coreference resolution, semantic role labeling, event classification, factuality, discourse and opinion. Each task is executed by one module, which allows to custom different pipeline topologies for text processing. The design of the Dutch, Italian and Spanish pipelines are also presented.

The design framework with the aim of analyzing the scaling capabilities of our NLP processing pipeline and the first experiments performed are also described in this document.

Contents

List of Tables

1 Introduction

This deliverable describes the first version of the **Event Detection** framework developed in NewsReader to process large and continuous streams of English, Dutch, Spanish and Italian news articles.

The research activities conducted within the NewsReader project strongly rely on the automatic detection of events, which are considered as the core information unit underlying news and therefore any decision making process that depends on news articles. The research focuses on four challenging aspects: event detection (addressed in WP04 -Event Detection-), event processing (addressed in WP05 -Event Modelling-), storage and reasoning over events (addressed in WP06 -Knowledge Store-), and scalling to large textual streams (addressed in WP2 -System Design-). Given that one of the main project goals is the extraction of event structures from large streams of documents and their management, a thorough analysis of what is an event, how its participants are characterized and how events are related to each other is of paramount importance.

WP04 (Event Detection) addresses the development of text processing modules that detect mentions of events, participants, their roles and the time and place expressions at a *document* level in the four project languages. An additional objective is to classify textual information on the factuality of the events and to derive the authority and trust of the source.

NewsReader uses an open and modular architecture for Natural Language Processing (NLP) as a starting point. The system uses the NLP Annotation Framework[1] (NAF) as a layered annotation format for text that can be shared across languages. NAF is an evolved version of KAF [Bosma *et al.*, 2009]. It includes more layers to represent additional linguistic phenomena and new ways to represent that information for the semantic web. Separate modules have been developed to add new interpretation layers using the output of previous layers. We developed new modules to perform event detection and to combine separate event representations. When necessary, new modules have been developed using the gold standards and training data (developed in WP03 -Benchmarking-). Specific input and output wrappers have been also developed or adapted to work with the new formats and APIs (also defined in WP02 -System Design-). For that, NewsReader exploited a variety of knowledge-rich and machine-learning approaches. During the first cycle of the project, we centered on English to provide the most advanced linguistic processing modules. But advanced NLP modules are being developed to cover similarly the other three languages of NewsReader. We are also completing advanced linguistic modules for the Spanish, Italian and Dutch as well as alternative English pipelines in order to tests the performance of different scaling infrastructures for advanced NLP processing. Thus, NewsReader also provides an abstraction layer for large-scale distributed computations, separating the "what" from the "how" of computation and isolating NLP developers from the details of concurrent programming.

Text-processing requires basic and generic NLP steps, such as tokenization, lemma-

[1] `https://github.com/ixa-ehu/NAF`

tization, part-of-speech tagging, parsing, word sense disambiguation, named entity and semantic role recognition for all the languages in NewsReader. Named entities are linked as much as possible to external sources such as Wikipedia and DBpedia. We use existing state-of-the-art technology and resources for this. Technology and resources have been selected for quality, efficiency, availability and extendability to other languages[2]. New techniques and resources are being developed for achieving interoperable semantic interpretation of English, Dutch, Spanish and Italian. Moreover, in subsequent cycles of the project, NewsReader will provide wide-coverage linguistic processors adapted to the financial domain.

The semantic interpretation of the text is directed towards the detection of event mentions and those named entities that play a role in these events, including time and location expressions. This implies covering all expressions (verbal, nominal and lexical) and meanings that can refer to events, their participating named entities, time and place expressions but also resolving any co-reference relations for these named entities and explicit (causal) relations between different event mentions. The analysis results in an augmentation of the text with semantic concepts and identifiers. This allows us to access lexical resources and ontologies that provide for each word and expression 1) the possible semantic type (e.g. to what type of event or participant it can refer), 2) the probability that it refers to that type (as scored by the word sense disambiguation and named entity recognition), 3) what types of participants are expected for each event (using background knowledge resources) and 4) what semantic roles are expected for each event (also using background knowledge resources. Such constraints are used by different rule-based, knowledge-rich and hybrid machine-learning systems to determine the actual event structures in texts.

We are also developing new classifiers that provide a factuality score which indicates the likelihood that an event took place (e.g. by exploiting textual and structural markers such as *not*, *failed*, *succeeded*, *might*, *should*, *will*, *probably*, etc.). Authority and trust will be derived from the metadata available on each source, the number of times the same information is expressed by different sources (combined with the type of source), but also on stylistic properties of the text (formal or informal, use of references, use of direct and indirect speech) and richness and coherence of the information that is given. For each unique event, we will also derive a trust and authority score based on the source data and a factuality score based on the textual properties. This information will be easily added to NAF in separate layers connected to each event.

The textual sources defined in WP01 (User Requirements) by the industrial partners come in various formats. In WP02 (System Design), we defined the RDF formats to represent the information of these sources. In WP04, we are processing the textual information to compatible RDF formats to make them available for subsequent NewsReader modules.

The remainder of the document consists of the following sections. Section 2 presents the event detection task designed for the first cycle of the NewsReader project. Section 3 presents the main NLP processing modules for English and the final three different

[2]See NewsReader deliverable D4.1 "Resources and linguistic processors"

English pipelines; IXA-pipeline[3], TextPro pipeline based on FBK TextPro[4] and CoreNLP pipeline based on Stanford CoreNLP[5]. Sections 4, 5 and 6 describe the Dutch, Italian and Spanish processing pipelines, respectively. Section 7 describe our initial plan for testing the performance of different scaling infrastructures for advanced NLP processing. Finally, Section 8 presents the main conclusions of this deliverable.

2 Event Detection

This section introduces the main NLP tasks addressed by the NewsReader project in order to process events across documents in four different languages: English, Dutch, Spanish and Italian. NewsReader Deliverable D4.1[6] provides a detailed survey about the current availability of resources and tools to perform event detection for the four languages involved in the project.

Event Detection (WP04) addresses the development of text processing modules that detect mentions of events, participants, their roles and the time and place expressions. Thus, text-processing requires basic and generic NLP steps, such as tokenization, lemmatization, part-of-speech tagging, parsing, word sense disambiguation, named entity and semantic role recognition for all the languages in NewsReader. Named entities are as much as possible linked to external sources (Wikipedia, DBpedia, JRC-Names, BabelNet, Freebase, etc.) and entity identifiers. Furthermore, event detection involves the identification of event mentions, event participants, the temporal constraints and, if relevant, the location. It also implies the detection of expressions of factuality of event mentions and the authority of the source of each event mention.

Moreover, NewsReader is developping:

- new techniques for achieving interoperable Semantic Interpretation of English, Dutch, Spanish and Italian
- wide-coverage linguistic processors adapted to the financial domain
- new scaling infrastructures for advanced NLP processing of large and continuous streams of English, Dutch, Spanish and Italian news articles.

During the first cycle of the NewsReader project (Event Detection, version 1) we focused on processing general English news. We considered three different advanced English pipelines:

- IXA-pipeline
- TextPro-based pipeline

[3] `https://github.com/ixa-ehu`
[4] `http://textpro.fbk.eu/`
[5] `http://nlp.stanford.edu/software/corenlp.shtml`
[6] `http://www.newsreader-project.eu/files/2012/12/NewsReader-316404-D4.1.pdf`

- Stanford-based CoreNLP pipeline

Finally, we decided to use the IXA-pipeline as a basic system for developing the English pipeline because it offers an open source, data-centric, modular, robust and efficient set of NLP tools for Spanish and English. It can be used "as is" or exploit its modularity to pick and change different components. Given its open-source nature, it can also be modified and extended for it to work with other languages.

IXA pipeline[7] provides *ready to use modules* to perform efficient and accurate linguistic annotation for English and Spanish[8]. More specifically, the objectives of IXA pipeline is to offer basic NLP technology that is:

1. **Simple and ready to use**: Every Java module of the IXA pipeline can be up an running after two simple steps.

2. **Portable**: The Java binaries are generated with "all batteries included" which means that no Java classpath configurations or installing of any third-party dependencies is required. The modules will run on any platform as long as a JVM 1.7+ is available.

3. **Modular**: Unlike other NLP toolkits, which often are built in a monolotic architecture, IXA pipeline is built in a data centric architecture so that modules can be picked and changed (even from other NLP toolkits). The modules behave like Unix pipes, they all take standard input, do some annotation, and produce standard output which in turn is the input for the next module.

4. **Efficient**: Piping the tokenizer, POS tagger and lemmatizer all in one process annotates over 5,500 words/second. The named-entity recognition module annotates over 5K words/second. In a multi-core machine, these times are dramatically reduced due to multi-threading (even 4 times faster). Furthermore, the most memory intensive process, the parser, requires less than 1GB of RAM.

5. **Multilingual**: Currently we offer NLP annotations for both English and Spanish, but other languages are being included in the pipeline.

6. **Accurate**: Previous points do not mean that IXA pipeline does not strive to offer accurate linguistic annotators. For example, POS tagging and NERC for English and Spanish are comparable with other state-of-the-art systems, as it is the coreference resolution module for English.

7. **Apache License 2.0**: IXA Pipeline is licensed under the Apache License 2.0, an open-source license that facilitates source code use, distribution and integration, also for commercial purposes.[9]

[7] `https://github.com/ixa-ehu`
[8] Modules for Italian, French, German and Dutch are also being developed as we speak.
[9] http://www.apache.org/licenses/LICENSE-2.0.html

IXA pipeline currently provides the following linguistic annotations: Sentence segmentation, tokenization, Part of Speech (POS) tagging, Lemmatization, Named Entity Recognition and Classification (NER), Syntactic Parsing and Coreference Resolution. To this basic pipeline we also have added new modules for Semantic Role Labelling, Named Entity Linling, TimeML annotations, event classification, factuality, discourse and opinion mining.

We will use the other pipelines to test the performance of different infrastructures, topologies and configurations for processing large and continuous streams of English news. During the second and third cycles of the project, we plan to extend the capabilities, coverage and precision of the English pipeline as well as the rest of languages covered in the project. For instance, in the second cycle of the NewsReader project, we also plan to adapt the English linguistic processors to the financial domain. In the third cycle of the project we also plan to adapt the rest of languages to the financial domain.

Although we will use NAF to harmonize the different outcomes, we also need to stablish a common semantic framework for representing the event mentions. For instance, SEMAFOR[10] uses FrameNet [Baker *et al.*, 1997] for semantic role labelling (SRL), while Mate-tools[11] uses PropBank [Palmer *et al.*, 2005] for the same task. Additionaly, as a backup solution, we are also processing the text with Word Sense Disambiguation modules to match WordNet [Fellbaum, 1998] identifiers across predicate models and languages.

To allow interoperable semantic interpretation of texts in multiple languages and predicate models, we started the development of the *Predicate Matrix*, a new lexical resource resulting from the integration of multiple sources of predicate information including FrameNet [Baker *et al.*, 1997], VerbNet [Kipper, 2005], PropBank [Palmer *et al.*, 2005] and WordNet [Fellbaum, 1998]. By using the Predicate Matrix, we expect to provide a more robust interoperable lexicon by discovering and solving inherent inconsistencies among the resources. We plan to extend the coverage of current predicate resources (by including from WordNet morphologically related nominal and verbal concepts), to enrich WordNet with predicate information, and possibly to extend predicate information to languages other than English (by exploiting the local wordnets aligned to the English WordNet).

2.1 Multilingual and Interoperable Predicate Models

Predicate models such as FrameNet [Baker *et al.*, 1997], VerbNet [Kipper, 2005] or PropBank [Palmer *et al.*, 2005] are core resources in most advanced NLP tasks such as Question Answering, Textual Entailment or Information Extraction. Most of the systems with Natural Language Understanding capabilities require a large and precise amount of semantic knowledge at the predicate-argument level. This type of knowledge allows to identify the underlying typical participants of a particular event independently of its realization in the text. Thus, using these models, different linguistic phenomena expressing the same event, such as active/passive transformations, verb alternations, nominalizations, implicit real-

[10] `http://code.google.com/p/semafor-semantic-parser/wiki/FrameNet`
[11] `http://code.google.com/p/mate-tools/`

izations can be harmonized into a common semantic representation. In fact, lately, several systems have been developed for shallow semantic parsing an explicit and implicit semantic role labeling using these resources [Erk and Pado, 2004], [Shi and Mihalcea, 2005], [Giuglea and Moschitti, 2006], [Laparra and Rigau, 2013].

However, building large and rich enough predicate models for broad–coverage semantic processing takes a great deal of expensive manual effort involving large research groups during long periods of development. In fact, the coverage of currently available predicate-argument resources is still far from complete. For example, [Burchardt *et al.*, 2005] or [Shen and Lapata, 2007] indicate the limited coverage of FrameNet as one of the main problems of this resource. In fact, FrameNet1.5 covers around 10,000 lexical-units while for instance, WordNet3.0 contains more than 150,000 words. Furthermore, the same effort should be invested for each different language [Subirats and Petruck, 2003].

Most previous research focuses on the integration of resources targeted at knowledge about nouns and named entities rather than predicate knowledge. Well known examples are YAGO [Suchanek *et al.*, 2007], Freebase [Bollacker *et al.*, 2008], DBPedia [Bizer *et al.*, 2009], BabelNet [Navigli and Ponzetto, 2010] or UBY [Gurevych *et al.*, 2012].

Following the line of previous works [Shi and Mihalcea, 2005], [Burchardt *et al.*, 2005], [Johansson and Nugues, 2007], [Pennacchiotti *et al.*, 2008], [Cao *et al.*, 2008], [Tonelli and Pianta, 2009], [Laparra *et al.*, 2010], we will also focus on the integration of predicate information. We start from the basis of SemLink [Palmer, 2009]. SemLink aimed to connect together different predicate resources such as FrameNet [Baker *et al.*, 1997], VerbNet [Kipper, 2005], PropBank [Palmer *et al.*, 2005] and WordNet [Fellbaum, 1998]. Although the mapping between the different sources of predicate information is far from complete, these resources can be combined in order to extend its coverage (by including from WordNet closely related nominal and verbal concepts), to discover inherent inconsistencies among the resources, to enrich WordNet with predicate information, and possibly to extend predicate information to languages other than English (by exploiting the local wordnets aligned to the English WordNet).

2.1.1 Sources of Predicate Information

Currently, we are considering the following sources of predicate information:

SemLink[12] [Palmer, 2009] is a project whose aim is to link together different predicate resources via set of mappings. These mappings makes it possible to combine the different information provided by these different lexical resources for tasks such as inferencing, consistency cheking, interoperable semantic role labelling, etc. We can also use this mappings to aid semi-automatic or fully automatic extensions of the current coverage of each of the resources, in order to increase the overall overlapping coverage. SemLink currently provides partial mappings between FrameNet [Baker *et al.*, 1997], VerbNet [Kipper, 2005], PropBank [Palmer *et al.*, 2005] and WordNet [Fellbaum, 1998].

[12] urlhttp://verbs.colorado.edu/semlink/

FrameNet [13] [Baker *et al.*, 1997] is a rich semantic resource that contains descriptions and corpus annotations of English words following the paradigm of Frame Semantics [Fillmore, 1976]. In frame semantics, a Frame corresponds to a scenario that involves the interaction of a set of typical participants, playing a particular role in the scenario. FrameNet groups words or lexical units (LUs hereinafter) into coherent semantic classes or frames, and each frame is further characterized by a list of participants or lexical elements (LEs hereinafter). Different senses for a word are represented in FrameNet by assigning different frames.

PropBank [14] [Palmer *et al.*, 2005] aims to provide a wide corpus annotated with information about semantic propositions, including relations between the predicates and their arguments. PropBank also contains a description of the frame structures, called framesets, of each sense of every verb that belong to its lexicon. Unlike other similar resources, as FrameNet, PropBank defines the arguments, or roles, of each verb individually. In consecuence, it becomes a hard task obtaning a generalization of the frame structures over the verbs.

VerbNet [15] [Kipper, 2005] hierarchical domain-independent broad-coverage verb lexicon for English. VerbNet is organized into verb classes extending [Levin, 1993] classes through refinement and addition of subclasses to achieve syntactic and semantic coherence among members of a class. Each verb class in VN is completely described by thematic roles, selectional restrictions on the arguments, and frames consisting of a syntactic description and semantic predicates.

WordNet [16] [Fellbaum, 1998] is by far the most widely-used knowledge base. In fact, WordNet is being used world-wide for anchoring different types of semantic knowledge including wordnets for languages other than English [Gonzalez-Agirre *et al.*, 2012a]. It contains manually coded information about English nouns, verbs, adjectives and adverbs and is organized around the notion of a *synset*. A synset is a set of words with the same part-of-speech that can be interchanged in a certain context. For example, <*learn*, *study*, *read*, *take*> form a synset because they can be used to refer to the same concept. A synset is often further described by a gloss, in this case: *"be a student of a certain subject"* and by explicit semantic relations to other synsets. Each synset represents a concept that are related with an large number of semantic relations, including hypernymy/hyponymy, meronymy/holonymy, antonymy, entailment, etc.

2.1.2 Extending the Predicate Matrix

Interestingly, the existing aligments also offer a very interesting source of information to be exploited. In fact, we are devising a number of simple automatic methods to extend SemLink by exploiting simple properties from WordNet. As a proof of concept, we present in this section two very simple approaches to extend the coverage of the mapping between

[13] `http://framenet.icsi.berkeley.edu/`
[14] `http://verbs.colorado.edu/~mpalmer/projects/ace.html`
[15] `http://verbs.colorado.edu/~mpalmer/projects/verbnet.html`
[16] `http://wordnet.princeton.edu/`

VerbNet predicates and WordNet senses. Moreover, we also plan to use additional semantic resources that use WordNet as a backbone. For instance, exploiting those knowledge resources integrated into the Multilingual Central Repository[17] (MCR) [Atserias *et al.*, 2004; Gonzalez-Agirre *et al.*, 2012a] to extend automatically the aligment of the different sources of predicate information (VerbNet, PropBank, FrameNet and WordNet). Following the line of previous works, in order to assign more WordNet verb senses to VerbNet predicates, we also plan to apply more sophisticated word-sense disambiguation algorithms to semantically coherent groups of predicates [Laparra *et al.*, 2010].

VerbNet Monosemous predicates Monosemous verbs from WordNet can be directly assigned to VerbNet predicates still without a WordNet aligment. This very simple strategy solves **240** aligments. In this way, VerbNet predicates such as $divulge_v$, $exhume_v$, $mutate_v$ or $upload_v$ obtain a corresponding WordNet word sense[18]. Note that only **576** lemmas from VerbNet are not aligned to WordNet.

WordNet synonyms A very straightforward method to extend the mapping between WordNet and VerbNet consists on including synonyms of already aligned WordNet senses as new members of the corresponding VerbNet class. Obviously, this method expects that WordNet synomyms share the same predicate information. For instance, the predicate $desert_v$ member of the VerbNet class leave-51.2-1 appears to be assigned to desert%2:31:00 WordNet verbal sense. In WordNet, this word sense also has three synonyms, $abandon_v$, $forsake_v$ and $desolate_v$. Obviously, the three verbal senses can also be assigned to the same VerbNet class. This simple approach can create up to **5,075** new members of VerbNet classes (corresponding to **4,616** different senses). For instance, Table 1 presents two productive examples. Moreover, **103** VerbNet predicates without mapping to WordNet in SemLink will be aligned to its corresponding WordNet sense.

VerbNet	WordNet	New
leave-51.2.1	desert%2:31:00	abandon%2:31:00:: forsake%2:31:00:: desolate%2:31:00::
remove-10.1	retract%2:32:00	abjure%2:32:00:: recant%2:32:00:: forswear%2:32:00:: resile%2:32:00::

Table 1: New WordNet senses aligned to VerbNet

2.1.3 Using WordNet to cross-check predicate information

As described in the previous section, it seems possible to apply simple but productive methods to extend the existing predicate aligments. In this section, we will also show as a

[17] `http://adimen.si.ehu.es/MCR`

[18] Obviously, these aligments can be considered just as sugestions to be revised later on manualy

proof-of-concept a simple way to exploit WordNet for validating the predicate information appearing in the lexicons. Again, we will apply a simple method to check the consistency of the VerbNet. Consider the following WordNet synset<*understand*, *read*, *interpret*, *translate*> with the gloss *"make sense of a language"* and the example sentences *"She understands French; Can you read Greek?"*. As synonyms, these verbs denote the same concept and are interchangeable in many contexts. In our initial versions of the Predicate Matrix *read%2:31:04* appears aligned with the VerbNet class learn-14-1[19] while one of its synonyms *understand%2:31:03* appears aligned to the VerbNet class comprehend-87.2[20]. Moreover, the thematic roles of both classes are different. Learn-14-1 has the following roles Agent [+animate], Topic and Source while comprehend-87.2 has Experiencer [+animate or +organization], Attribute and Stimulus. Are both sets of roles compatible? Complementary? Is one of them incorrect? Should we joint them? Maybe is the aligment incorrect? Is perhaps the synset definition?

Following with this example, the VerbNet predicate $understand_v$ has no connection to FrameNet, but its VerbNet class comprehend-87.2-1 has some other verbal predicates aligned to FrameNet. For instance, $apprehend_v$, $comprehend_v$ and $grasp_v$ are linked to the Grasp[21] FrameNet frame. Among the lexical units corresponding to the Grasp frame it appears also the verbal predicate $understand_v$. This means that possibly, this Verbal predicate should also be aligned to the FrameNet frame Grasp. The core lexical elements (roles) of this frame are Cognizer with semantic type Sentient, Faculty and Phenomenon.

We are now producing and studying the initial versions of the Predicate Matrix by exploiting SemLink and applying very simple methods to extend and validate its content. Table 2 presents a full example of the information that is currently available in the Predicate Matrix. Each row of this table represents the mapping of a role over the different resources and includes all the aligned knowledge about its corresponding verb. The table presents the cases obtained originally from SemLink, denoted as *OLD*, and the cases inferred following the methods explained previously, identified as *NEW*. The table also includes the following fields: the lemma in VerbNet, the sense of the verb in WordNet, the thematic role in VerbNet, the Frame of FrameNet, the corresponding FrameElement of FrameNet, the predicate in PropBank and its argument, the offset of the sense in WordNet and the knowledge associated with that sense in the MCR, such as its Base Concept [Izquierdo *et al.*, 2007], new WordNet domain aligned to WordNet 3.0 [González-Agirre *et al.*, 2012b], Adimen-SUMO [Álvez *et al.*, 2012] and EuroWordNet Top Ontology [Álvez *et al.*, 2008] features. Finaly, its line also includes the frequency and the number of relations of the WordNet sense.

2.1.4 Towards the Predicate Matrix

By using the Predicate Matrix, we expect to provide a more robust interoperable lexicon. We plan to discover and solve inherent inconsistencies among the integrated resources.

[19] `http://verbs.colorado.edu/verb-index/vn/learn-14.php#learn-14-1`

[20] `http://verbs.colorado.edu/verb-index/vn/comprehend-87.2.php#comprehend-87.2-1`

[21] `https://framenet2.icsi.berkeley.edu/fnReports/data/frame/Grasp.xml`

We plan to extend the coverage of current predicate resources (by including from WordNet morphologically related nominal and verbal concepts, by exploiting also FrameNet information, etc.), to enrich WordNet with predicate information, and possibly to extend predicate information to languages other than English (by exploiting the local wordnets aligned to the English WordNet) and predicate information from other languages. With this purpose we plan to use resourches such as Ancora [Taulé *et al.*, 2008].

3 English NLP Processing

As previously Section 1, during the first cycle of the project, we center mainly on English when implementing the linguistic processing modules. This section describes the developed text processing modules for event detection that detect mentions of events, participants, their roles and the time and place expressions in English. The modules dealing with more basic NLP steps such as tokenization, lemmatization etc. are also presented. Finally, modules to perform event detection, factuality, discourse and opinion are described.

Sections 3.1-3.13 presents the developed modules. The descriptions of the modules are provided along with some technical information: input layer(s); output layer(s); required modules; level of operation and language dependent. Section 3.14 presents a graphical representation of the various pipelines.

3.1 Tokenizer

The tokenization can be performed by means of three different modules: Ixa-pipe tokenizer, Stanford-based tokenizer and TokenPro-based tokenizer. The Ixa-pipe tokenizer is the one integrated in the prototype.

3.1.1 Ixa-pipe Tokenizer

- Module: M1.1
- Description of the module: This module provides Sentence Segmentation and Tokenization for English and Spanish via two methods: a) Rule-based approach originally inspired by the Moses[22] tokenizer but with several additions and modifications. These include treatment for URLs, multi punctuation marks for start and end of sentences, and more comprehensive gazeteers of expressions that need not be tokenized or splitted. This is the default; and b) Probabilistic models trained using the Apache OpenNLP API [23] based on the CoNLL 2003 and 2002 datasets. For English we also offer Penn Treebank style of tokenization [Marcus *et al.*, 1993], which is for example useful for syntactic parsing. Similarly, the Spanish tokenizer is implemented to be compatible with the Ancora corpus [Taulé *et al.*, 2008]. The module is part

[22] `https://github.com/moses-smt/mosesdecoder`
[23] `http://opennlp.apache.org/`

VERSION	VN_LEMA	WN_SENSE	VN_THEMROLE	FN_FRAME	FN_LEXENT	FN_ROLE	PB_ROLESET	PB_ARG	MCR_iliOffset	MCR_BC	MCR_DOMAIN	MCR_SUMO	MCR_TO	MCR_LEXNAME	WN_BLC	WN_SENSEFREC	WN_SYNSET_REL_NUM
OLD	allow	allow%2:32:00	Agent	Grant_permission	NULL	Grantor	allow.01	0	ili-30-00802318-v	1	factotum	IntentionalProcess	Agentive;BoundedEvent;Communication;Purpose;Social;	communication	act%2:41:00	49	021
OLD	allow	allow%2:32:00	Agent	Permitting	8066	Principle	allow.01	0	ili-30-00802318-v	1	factotum	IntentionalProcess	Agentive;BoundedEvent;Communication;Purpose;Social;	communication	act%2:41:00	49	021
OLD	allow	allow%2:32:00	Theme	Grant_permission	NULL	Grantee	allow.01	NULL	ili-30-00802318-v	1	factotum	IntentionalProcess	Agentive;BoundedEvent;Communication;Purpose;Social;	communication	act%2:41:00	49	021
OLD	allow	allow%2:32:00	Theme	Permitting	8066	State_of_affairs	allow.01	NULL	ili-30-00802318-v	1	factotum	IntentionalProcess	Agentive;BoundedEvent;Communication;Purpose;Social;	communication	act%2:41:00	49	021
OLD	allow	allow%2:32:00	Location	Grant_permission	NULL	Action	allow.01	1	ili-30-00802318-v	1	factotum	IntentionalProcess	Agentive;BoundedEvent;Communication;Purpose;Social;	communication	act%2:41:00	49	021
OLD	allow	allow%2:32:00	Location	Grant_permission	NULL	Place	allow.01	1	ili-30-00802318-v	1	factotum	IntentionalProcess	Agentive;BoundedEvent;Communication;Purpose;Social;	communication	act%2:41:00	49	021
OLD	allow	allow%2:32:00	Location	Permitting	8066	State_of_affairs	allow.01	1	ili-30-00802318-v	1	factotum	IntentionalProcess	Agentive;BoundedEvent;Communication;Purpose;Social;	communication	act%2:41:00	49	021
NEW	countenance	countenance%2:32:00	Agent	Grant_permission	NULL	Grantor	allow.01	0	ili-30-00802318-v	1	factotum	IntentionalProcess	Agentive;BoundedEvent;Communication;Purpose;Social;	communication	act%2:41:00	49	021
NEW	countenance	countenance%2:32:00	Agent	Permitting	8066	Principle	allow.01	0	ili-30-00802318-v	1	factotum	IntentionalProcess	Agentive;BoundedEvent;Communication;Purpose;Social;	communication	act%2:41:00	49	021
NEW	countenance	countenance%2:32:00	Theme	Grant_permission	NULL	Grantee	allow.01	NULL	ili-30-00802318-v	1	factotum	IntentionalProcess	Agentive;BoundedEvent;Communication;Purpose;Social;	communication	act%2:41:00	49	021
NEW	countenance	countenance%2:32:00	Theme	Permitting	8066	State_of_affairs	allow.01	NULL	ili-30-00802318-v	1	factotum	IntentionalProcess	Agentive;BoundedEvent;Communication;Purpose;Social;	communication	act%2:41:00	49	021
NEW	countenance	countenance%2:32:00	Location	Grant_permission	NULL	Action	allow.01	1	ili-30-00802318-v	1	factotum	IntentionalProcess	Agentive;BoundedEvent;Communication;Purpose;Social;	communication	act%2:41:00	49	021
NEW	countenance	countenance%2:32:00	Location	Grant_permission	NULL	Place	allow.01	1	ili-30-00802318-v	1	factotum	IntentionalProcess	Agentive;BoundedEvent;Communication;Purpose;Social;	communication	act%2:41:00	49	021
NEW	countenance	countenance%2:32:00	Location	Permitting	8066	State_of_affairs	allow.01	1	ili-30-00802318-v	1	factotum	IntentionalProcess	Agentive;BoundedEvent;Communication;Purpose;Social;	communication	act%2:41:00	49	021
NEW	let	let%2:32:02	Agent	Grant_permission	NULL	Grantor	allow.01	0	ili-30-00802318-v	1	factotum	IntentionalProcess	Agentive;BoundedEvent;Communication;Purpose;Social;	communication	act%2:41:00	49	021
NEW	let	let%2:32:02	Agent	Permitting	8066	Principle	allow.01	0	ili-30-00802318-v	1	factotum	IntentionalProcess	Agentive;BoundedEvent;Communication;Purpose;Social;	communication	act%2:41:00	49	021
NEW	let	let%2:32:02	Theme	Grant_permission	NULL	Grantee	allow.01	NULL	ili-30-00802318-v	1	factotum	IntentionalProcess	Agentive;BoundedEvent;Communication;Purpose;Social;	communication	act%2:41:00	49	021
NEW	let	let%2:32:02	Theme	Permitting	8066	State_of_affairs	allow.01	NULL	ili-30-00802318-v	1	factotum	IntentionalProcess	Agentive;BoundedEvent;Communication;Purpose;Social;	communication	act%2:41:00	49	021
NEW	let	let%2:32:02	Location	Grant_permission	NULL	Action	allow.01	1	ili-30-00802318-v	1	factotum	IntentionalProcess	Agentive;BoundedEvent;Communication;Purpose;Social;	communication	act%2:41:00	49	021
NEW	let	let%2:32:02	Location	Grant_permission	NULL	Place	allow.01	1	ili-30-00802318-v	1	factotum	IntentionalProcess	Agentive;BoundedEvent;Communication;Purpose;Social;	communication	act%2:41:00	49	021
NEW	let	let%2:32:02	Location	Permitting	8066	State_of_affairs	allow.01	1	ili-30-00802318-v	1	factotum	IntentionalProcess	Agentive;BoundedEvent;Communication;Purpose;Social;	communication	act%2:41:00	49	021
OLD	permit	permit%2:32:00	Agent	Grant_permission	8024	Grantor	permit.01	0	ili-30-00802318-v	1	factotum	IntentionalProcess	Agentive;BoundedEvent;Communication;Purpose;Social;	communication	act%2:41:00	55	021
OLD	permit	permit%2:32:00	Agent	Grant_permission	8024	Grantor	permit.01	2	ili-30-00802318-v	1	factotum	IntentionalProcess	Agentive;BoundedEvent;Communication;Purpose;Social;	communication	act%2:41:00	55	021
OLD	permit	permit%2:32:00	Agent	Permitting	8450	Principle	permit.01	0	ili-30-00802318-v	1	factotum	IntentionalProcess	Agentive;BoundedEvent;Communication;Purpose;Social;	communication	act%2:41:00	55	021
OLD	permit	permit%2:32:00	Agent	Permitting	8450	Principle	permit.01	2	ili-30-00802318-v	1	factotum	IntentionalProcess	Agentive;BoundedEvent;Communication;Purpose;Social;	communication	act%2:41:00	55	021
OLD	permit	permit%2:32:00	Theme	Grant_permission	8024	Grantee	permit.01	NULL	ili-30-00802318-v	1	factotum	IntentionalProcess	Agentive;BoundedEvent;Communication;Purpose;Social;	communication	act%2:41:00	55	021
OLD	permit	permit%2:32:00	Theme	Permitting	8450	State_of_affairs	permit.01	NULL	ili-30-00802318-v	1	factotum	IntentionalProcess	Agentive;BoundedEvent;Communication;Purpose;Social;	communication	act%2:41:00	55	021
OLD	permit	permit%2:32:00	Location	Grant_permission	8024	Action	permit.01	1	ili-30-00802318-v	1	factotum	IntentionalProcess	Agentive;BoundedEvent;Communication;Purpose;Social;	communication	act%2:41:00	55	021
OLD	permit	permit%2:32:00	Location	Grant_permission	8024	Place	permit.01	1	ili-30-00802318-v	1	factotum	IntentionalProcess	Agentive;BoundedEvent;Communication;Purpose;Social;	communication	act%2:41:00	55	021
OLD	permit	permit%2:32:00	Location	Permitting	8450	State_of_affairs	permit.01	1	ili-30-00802318-v	1	factotum	IntentionalProcess	Agentive;BoundedEvent;Communication;Purpose;Social;	communication	act%2:41:00	55	021
OLD	admit	admit%2:41:01	Agent	NULL	NULL	NULL	admit.02	0	ili-30-02502536-v	0	factotum	confersRight	Agentive;BoundedEvent;Communication;Purpose;Social;	social	permit%2:32:00	11	010
OLD	admit	admit%2:41:01	Theme	NULL	NULL	NULL	admit.02	1	ili-30-02502536-v	0	factotum	confersRight	Agentive;BoundedEvent;Communication;Purpose;Social;	social	permit%2:32:00	11	010
OLD	admit	admit%2:41:01	Location	NULL	NULL	NULL	admit.02	2	ili-30-02502536-v	0	factotum	confersRight	Agentive;BoundedEvent;Communication;Purpose;Social;	social	permit%2:32:00	11	010
NEW	allow_in	allow_in%2:41:00	Agent	NULL	NULL	NULL	admit.02	0	ili-30-02502536-v	0	factotum	confersRight	Agentive;BoundedEvent;Communication;Purpose;Social;	social	permit%2:32:00	11	010
NEW	allow_in	allow_in%2:41:00	Theme	NULL	NULL	NULL	admit.02	1	ili-30-02502536-v	0	factotum	confersRight	Agentive;BoundedEvent;Communication;Purpose;Social;	social	permit%2:32:00	11	010
NEW	allow_in	allow_in%2:41:00	Location	NULL	NULL	NULL	admit.02	2	ili-30-02502536-v	0	factotum	confersRight	Agentive;BoundedEvent;Communication;Purpose;Social;	social	permit%2:32:00	11	010
NEW	intromit	intromit%2:41:00	Agent	NULL	NULL	NULL	admit.02	0	ili-30-02502536-v	0	factotum	confersRight	Agentive;BoundedEvent;Communication;Purpose;Social;	social	permit%2:32:00	11	010
NEW	intromit	intromit%2:41:00	Theme	NULL	NULL	NULL	admit.02	1	ili-30-02502536-v	0	factotum	confersRight	Agentive;BoundedEvent;Communication;Purpose;Social;	social	permit%2:32:00	11	010
NEW	intromit	intromit%2:41:00	Location	NULL	NULL	NULL	admit.02	2	ili-30-02502536-v	0	factotum	confersRight	Agentive;BoundedEvent;Communication;Purpose;Social;	social	permit%2:32:00	11	010
NEW	let_in	let_in%2:41:02	Agent	NULL	NULL	NULL	admit.02	0	ili-30-02502536-v	0	factotum	confersRight	Agentive;BoundedEvent;Communication;Purpose;Social;	social	permit%2:32:00	11	010
NEW	let_in	let_in%2:41:02	Theme	NULL	NULL	NULL	admit.02	1	ili-30-02502536-v	0	factotum	confersRight	Agentive;BoundedEvent;Communication;Purpose;Social;	social	permit%2:32:00	11	010
NEW	let_in	let_in%2:41:02	Location	NULL	NULL	NULL	admit.02	2	ili-30-02502536-v	0	factotum	confersRight	Agentive;BoundedEvent;Communication;Purpose;Social;	social	permit%2:32:00	11	010
OLD	admit	admit%2:41:00	Agent	NULL	NULL	NULL	admit.02	0	ili-30-02449847-v	0	factotum	confersRight	Agentive;BoundedEvent;Communication;Purpose;Social;	social	permit%2:32:00	1	008
OLD	admit	admit%2:41:00	Theme	NULL	NULL	NULL	admit.02	1	ili-30-02449847-v	0	factotum	confersRight	Agentive;BoundedEvent;Communication;Purpose;Social;	social	permit%2:32:00	1	008
OLD	admit	admit%2:41:00	Location	NULL	NULL	NULL	admit.02	2	ili-30-02449847-v	0	factotum	confersRight	Agentive;BoundedEvent;Communication;Purpose;Social;	social	permit%2:32:00	1	008
OLD	include	include%2:41:03	Agent	NULL	NULL	NULL	include.01	0	ili-30-02449847-v	0	factotum	confersRight	Agentive;BoundedEvent;Communication;Purpose;Social;	social	permit%2:32:00	8	008
OLD	include	include%2:41:03	Theme	NULL	NULL	NULL	include.01	1	ili-30-02449847-v	0	factotum	confersRight	Agentive;BoundedEvent;Communication;Purpose;Social;	social	permit%2:32:00	8	008
OLD	include	include%2:41:03	Location	NULL	NULL	NULL	include.01	2	ili-30-02449847-v	0	factotum	confersRight	Agentive;BoundedEvent;Communication;Purpose;Social;	social	permit%2:32:00	8	008
NEW	let_in	let_in%2:41:00	Agent	NULL	NULL	NULL	admit.02	0	ili-30-02449847-v	0	factotum	confersRight	Agentive;BoundedEvent;Communication;Purpose;Social;	social	permit%2:32:00	1	008
NEW	let_in	let_in%2:41:00	Theme	NULL	NULL	NULL	admit.02	1	ili-30-02449847-v	0	factotum	confersRight	Agentive;BoundedEvent;Communication;Purpose;Social;	social	permit%2:32:00	1	008
NEW	let_in	let_in%2:41:00	Location	NULL	NULL	NULL	admit.02	2	ili-30-02449847-v	0	factotum	confersRight	Agentive;BoundedEvent;Communication;Purpose;Social;	social	permit%2:32:00	1	008
NEW	except	except%2:31:00	Agent	Inclusion	NULL	NULL	NULL	NULL	ili-30-00615774-v	0	buildings	IntentionalProcess	BoundedEvent;Existence;	cognition	destroy%2:36:00	15	008
NEW	except	except%2:31:00	Theme	Inclusion	NULL	NULL	NULL	NULL	ili-30-00615774-v	0	buildings	IntentionalProcess	BoundedEvent;Existence;	cognition	destroy%2:36:00	15	008
NEW	except	except%2:31:00	Location	Inclusion	NULL	NULL	NULL	NULL	ili-30-00615774-v	0	buildings	IntentionalProcess	BoundedEvent;Existence;	cognition	destroy%2:36:00	15	008
OLD	exclude	exclude%2:31:01	Agent	Inclusion	NULL	NULL	NULL	NULL	ili-30-00615774-v	0	buildings	IntentionalProcess	BoundedEvent;Existence;	cognition	destroy%2:36:00	15	008
OLD	exclude	exclude%2:31:01	Theme	Inclusion	NULL	NULL	NULL	NULL	ili-30-00615774-v	0	buildings	IntentionalProcess	BoundedEvent;Existence;	cognition	destroy%2:36:00	15	008
OLD	exclude	exclude%2:31:01	Location	Inclusion	NULL	NULL	NULL	NULL	ili-30-00615774-v	0	buildings	IntentionalProcess	BoundedEvent;Existence;	cognition	destroy%2:36:00	15	008
NEW	leave_off	leave_off%2:31:00	Agent	Inclusion	NULL	NULL	NULL	NULL	ili-30-00615774-v	0	buildings	IntentionalProcess	BoundedEvent;Existence;	cognition	destroy%2:36:00	15	008
NEW	leave_off	leave_off%2:31:00	Theme	Inclusion	NULL	NULL	NULL	NULL	ili-30-00615774-v	0	buildings	IntentionalProcess	BoundedEvent;Existence;	cognition	destroy%2:36:00	15	008
NEW	leave_off	leave_off%2:31:00	Location	Inclusion	NULL	NULL	NULL	NULL	ili-30-00615774-v	0	buildings	IntentionalProcess	BoundedEvent;Existence;	cognition	destroy%2:36:00	15	008
NEW	leave_out	leave_out%2:31:01	Agent	Inclusion	NULL	NULL	NULL	NULL	ili-30-00615774-v	0	buildings	IntentionalProcess	BoundedEvent;Existence;	cognition	destroy%2:36:00	15	008
NEW	leave_out	leave_out%2:31:01	Theme	Inclusion	NULL	NULL	NULL	NULL	ili-30-00615774-v	0	buildings	IntentionalProcess	BoundedEvent;Existence;	cognition	destroy%2:36:00	15	008
NEW	leave_out	leave_out%2:31:01	Location	Inclusion	NULL	NULL	NULL	NULL	ili-30-00615774-v	0	buildings	IntentionalProcess	BoundedEvent;Existence;	cognition	destroy%2:36:00	15	008
NEW	take_out	take_out%2:31:00	Agent	Inclusion	NULL	NULL	NULL	NULL	ili-30-00615774-v	0	buildings	IntentionalProcess	BoundedEvent;Existence;	cognition	destroy%2:36:00	15	008
NEW	take_out	take_out%2:31:00	Theme	Inclusion	NULL	NULL	NULL	NULL	ili-30-00615774-v	0	buildings	IntentionalProcess	BoundedEvent;Existence;	cognition	destroy%2:36:00	15	008
NEW	take_out	take_out%2:31:00	Location	Inclusion	NULL	NULL	NULL	NULL	ili-30-00615774-v	0	buildings	IntentionalProcess	BoundedEvent;Existence;	cognition	destroy%2:36:00	15	008
OLD	exclude	exclude%2:41:00	Agent	Inclusion	NULL	NULL	NULL	NULL	ili-30-02449340-v	0	factotum	prevents	BoundedEvent;Cause;	social	prevent%2:41:00	7	007
OLD	exclude	exclude%2:41:00	Theme	Inclusion	NULL	NULL	NULL	NULL	ili-30-02449340-v	0	factotum	prevents	BoundedEvent;Cause;	social	prevent%2:41:00	7	007
OLD	exclude	exclude%2:41:00	Location	Inclusion	NULL	NULL	NULL	NULL	ili-30-02449340-v	0	factotum	prevents	BoundedEvent;Cause;	social	prevent%2:41:00	7	007
NEW	keep_out	keep_out%2:41:00	Agent	Inclusion	NULL	NULL	NULL	NULL	ili-30-02449340-v	0	factotum	prevents	BoundedEvent;Cause;	social	prevent%2:41:00	7	007
NEW	keep_out	keep_out%2:41:00	Theme	Inclusion	NULL	NULL	NULL	NULL	ili-30-02449340-v	0	factotum	prevents	BoundedEvent;Cause;	social	prevent%2:41:00	7	007
NEW	keep_out	keep_out%2:41:00	Location	Inclusion	NULL	NULL	NULL	NULL	ili-30-02449340-v	0	factotum	prevents	BoundedEvent;Cause;	social	prevent%2:41:00	7	007
NEW	shut	shut%2:41:00	Agent	Inclusion	NULL	NULL	NULL	NULL	ili-30-02449340-v	0	factotum	prevents	BoundedEvent;Cause;	social	prevent%2:41:00	7	007
NEW	shut	shut%2:41:00	Theme	Inclusion	NULL	NULL	NULL	NULL	ili-30-02449340-v	0	factotum	prevents	BoundedEvent;Cause;	social	prevent%2:41:00	7	007
NEW	shut	shut%2:41:00	Location	Inclusion	NULL	NULL	NULL	NULL	ili-30-02449340-v	0	factotum	prevents	BoundedEvent;Cause;	social	prevent%2:41:00	7	007
NEW	shut_out	shut_out%2:41:00	Agent	Inclusion	NULL	NULL	NULL	NULL	ili-30-02449340-v	0	factotum	prevents	BoundedEvent;Cause;	social	prevent%2:41:00	7	007
NEW	shut_out	shut_out%2:41:00	Theme	Inclusion	NULL	NULL	NULL	NULL	ili-30-02449340-v	0	factotum	prevents	BoundedEvent;Cause;	social	prevent%2:41:00	7	007
NEW	shut_out	shut_out%2:41:00	Location	Inclusion	NULL	NULL	NULL	NULL	ili-30-02449340-v	0	factotum	prevents	BoundedEvent;Cause;	social	prevent%2:41:00	7	007
OLD	welcome	welcome%2:35:00	Agent	NULL	NULL	NULL	welcome.01	0	ili-30-01470098-v	0	factotum	Communication	Dynamic;Location;	contact	receive%2:35:00	2	002
OLD	welcome	welcome%2:35:00	Theme	NULL	NULL	NULL	welcome.01	1	ili-30-01470098-v	0	factotum	Communication	Dynamic;Location;	contact	receive%2:35:00	2	002
OLD	welcome	welcome%2:35:00	Location	NULL	NULL	NULL	welcome.01	2	ili-30-01470098-v	0	factotum	Communication	Dynamic;Location;	contact	receive%2:35:00	2	002

Table 2: Partial view of the current content of the Predicate Matrix

of IXA-Pipeline ("is a pipeline"), a multilingual NLP pipeline developed by the IXA NLP Group. The module accepts raw text as standard input and outputs tokenized text in NAF format.

- Input layer(s): raw text
- Output layer(s): Tokens, sentences
- Required modules: –
- Level of operation: document level
- Language dependent: yes

3.1.2 Stanford-based Tokenizer

- Module: M1.2
- Description of the module: This module provides Sentence Segmentation and Tokenization for English as provided in the Stanford CoreNLP suite.[24]. The module performs the processes based on PTBTokenizer, a deterministic tokenizer implemented as a finite automaton. The output of the process is represented in NAF format.
- Input layer(s): raw text
- Output layer(s): Tokens, sentences
- Required modules: –
- Level of operation: document level
- Language dependent: yes

3.1.3 TokenPro-based Tokenizer

- Module: M1.3
- Description of the module: This module corresponds to TokenPro tokenizer. TokenPro is part of TextPro. TextPro[25] is a flexible, customizable, integratable and easy-to-use NLP tool, which has a set of modules to process raw or customized text and perform NLP tasks such as: web page cleaning, tokenization, sentence detection, morphological analysis, pos-tagging, lemmatization, chunking and named-entity recognition. The current version, TextPro 2.0, supports English and Italian languages.

[24] `http://nlp.stanford.edu/software/corenlp.shtml`
[25] `http://textpro.fbk.eu/`

TokenPro is a rule-based splitter to tokenize raw text, using some predefined rules specific for each language and producing one token per line. Tokenization can be quickly customized by editing specific splitting word rules or handling UTF-8 common/uncommon symbols, such as apostrophe, quote, dash, ecc. according with their usage in the domain. TokenPro provides also: a) UTF8 normalization of the token; b) the char position of the token inside the input text; c) sentence splitting. It obtains 98% accuracy.

- Input layer(s): raw text
- Output layer(s): Tokens, normalized tokens, char position of the tokens, sentences
- Required modules: –
- Level of operation: document level
- Language dependent: yes

3.2 POS-tagger

The part of speech tagging can be performed by means of three different modules: Ixa-pipe POS tagger, Stanford-based POS tagger and TextPro-based POS tagger. The Ixa-pipe POS tagger is the integrated one in the prototype.

3.2.1 Ixa-pipe POS tagger

- Module: M2.1
- Description of the module: This module provides POS tagging and lemmatization for English and Spanish. This module is part of IXA-Pipeline ("is a pipeline"), a multilingual NLP pipeline developed by the IXA NLP Group POS tagging models have been trained using the Apache OpenNLP API.[26] English perceptron models have been trained and evaluated using the WSJ treebank as explained in [Toutanova *et al.*, 2003]. Currently we obtain a performance of 96.48% vs 97.24% obtained by [Toutanova *et al.*, 2003]. Lemmatization is dictionary based and for English WordNet-3.0 is used. It is possible to use two dictionaries: a) plain text dictionary: en-lemmas.dict is a "Word POStag lemma" dictionary in plain text to perform lemmatization; b) Morfologik-stemming: english.dict is the same as en-lemmas.dict but binarized as a finite state automata using the morfologik-stemming project. This method uses 10% of RAM with respect to the plain text dictionary and works noticeably faster. By default lemmatization is performed using the Morfologik-stemming[27] binary dictionaries. The module accepts tokenized text in NAF format as standard input and outputs NAF.

[26] `http://opennlp.apache.org/`
[27] `http://sourceforge.net/projects/morfologik/`

- Input layer(s): Tokens
- Output layer(s): Lemmas, POS-tags
- Required modules: Tokenizer module
- Level of operation: Sentence level
- Language dependent: yes

3.2.2 Stanford-based POS tagger

- Module: M2.2
- Description of the module: This module is based on the java implementation of the Stanford Part-Of-Speech Tagger. It is an implementation of the log-linear POS tagger described in [Toutanova *et al.*, 2003]. This English tagger uses the Peen Treebank tag set and it has been trained on the WSJ treebank using the left3words architecture. The input and output of the process is represented in NAF format.
- Input layer(s): Tokens
- Output layer(s): Lemmas, POS-tags
- Required modules: Tokenizer module
- Level of operation: Sentence level
- Language dependent: yes

3.2.3 TextPro-based POS tagger

- Module: M2.3
- Description of the module: This module performs three processes: morphological analysis, PoS-Tagging and lemmatization. MorphoPro, the morphological analyzer, produces all possible morphological analyses for each token. The English model is trained using the BNC tagset. TagPro processes the tokens to assign them their part of speech. Given the morphological analyses and the PoS tag, the lemmatizer (LemmaPro) selects the compatible morphological analysis and the lemma of a token. TextPro is part of TextPro.
- Input layer(s): Tokens
- Output layer(s): POS-tags, lemmas
- Required modules: Tokenizer

- Level of operation: Sentence level
- Language dependent: yes

3.3 Parser

The event classifier includes a constituency parser and a dependency parser. The constituents can be obtained by means of the Ixa-pipe parser and the ChunkPro-based parser. The integrated one in the prototype is the Ixa-pipe parser. The dependencies are provided by the Mate-tools based parser and the Stanford-based parser. The integrated one is the Mate-tools based parser.

3.3.1 Ixa-pipe Parser

- Module: M3.1
- Description of the module: This module uses Apache OpenNLP programatically to provide Constituent Parsing for English and Spanish. English models have been trained using Penn treebank. This module is part of IXA-Pipeline ("is a pipeline"), a multilingual NLP pipeline developed by the IXA NLP Group. The module accepts lemmatized and POS tagged text in NAF format as standard input and outputs NAF.
- Input layer(s): lemma, POS
- Output layer(s): constituents
- Required modules: tokenizer, POS-tagger modules
- Level of operation: Sentence level
- Language dependent: yes

3.3.2 Mate-tools based Parser

- Module: M3.2
- Description of the module: This module is based on the MATE-tools [Björkelund *et al.*, 2010], a pipeline of linguistic processors that performs lemmatization, part-of-speech tagging, dependency parsing, and semantic role labeling of a sentence. As the input of the module is a NAF file that includes lemmatization and pos-tagging, the module only implements the dependency parser [Bohnet, 2010]. The module is ready to work with Spanish and English. For the latter one, the dependency parser had the top score in the CoNLL shared task 2009, obtaining 90.24% Labeled attachment score (LAS).

- Input layer(s): lemma, POS
- Output layer(s): dependencies
- Required modules: tokenizer, POS-tagger modules
- Level of operation: Sentence level
- Language dependent: yes

3.3.3 Stanford-based Parser

- Module: M3.3
- Description of the module: This module is based on the java implementation of the Stanford statistical parser. It is an implementation of the probabilistic lexicalized dependency parser [Klein and Manning, 2003]. The parser implements a factored product model using an A* algorithm. The input and output of the process is represented in NAF format.
- Input layer(s): lemma, POS
- Output layer(s): Stanford dependencies, phrase structure trees
- Required modules: tokenizer, POS-tagger modules
- Level of operation: Sentence level
- Language dependent: yes

3.3.4 ChunkPro-based Parser

- Module: M3.4
- Description of the module: ChunkPro is a chunker that groups words into flat constituents. This module uses a machine learning approach trained on CoNLL-2000 for English (95.28% accuracy). It uses the following chunk categories: ADJP (adjectival phrase), ADVP (adverbal phrase), CONJP (conjunction phrase), INTJ (interjection), LST (list marker, includes surrounding punctuation), NP (noun phrase), PP (prepositional phrase), PRT (particle), B-SBAR (clause introduced by a, possibly empty, subordinating conjunction), UCP (unlike coordinated phrase), VP (verb phrase). Output chunks are represented in the IOB2 format. ChunkPro is part of TextPro.
- Input layer(s): Tokens, POS-tags
- Output layer(s): Chunks
- Required modules: Tokenizer, POS-tagger

- Level of operation: Sentence level
- Language dependent: yes

3.4 Time Expressions

- Module: M4.1
- Description of the module: temporal expressions recognizer, TimePro: it identifies the tokens corresponding to temporal expressions in English and assignes them to one of the 4 Timex classes defined in ISO-TimeML. It is based on machine learning and it is trained on TempEval3 data. The average result for English is: 83.81% precision, 75.94% recall and 79.61% F1-measure values. This module has been integrated into TextPro pipeline but it also accepts a NAF input file with tokenization, POS-tagging and chunking information and provides a NAF file as output.
- Input layer(s): Tokens, POS tags, Chunks
- Output layer(s): Timex3
- Required modules: tokenizer, POS-tagger, Chunker
- Level of operation: sentence level, document level
- Language dependent: yes

3.5 Named Entity Recognizer

The named entities are provided by two different modules: the Ixa-pipe Named Entity Recognizer and the EntityPro-based Named Entity Recognizer. The Ixa-pipe Named Entity Recognizer is the integrated one in the prototype.

3.5.1 Ixa-pipe Named Entity Recognizer

- Module: M5.1
- Description of the module: This module uses Apache OpenNLP programatically to perform Named Entity Recognitionfor English and Spanish. English models have been trained using CoNLL 2003 dataset (84.80 F1). For Spanish, we obtain best results training Maximum Entropy models on the CoNLL 2002 dataset (79.92 F1). This module is part of IXA-Pipeline ("is a pipeline"), a multilingual NLP pipeline developed by the IXA NLP Group. The module accepts lemmatized and POS tagged text in NAF format as standard input and outputs a modified NAF (with named entities).
- Input layer(s): lemma, POS

- Output layer(s): named entities
- Required modules: POS-tagger module
- Level of operation: sentence level
- Language dependent: yes

3.5.2 EntityPro-based Named Entity Recognizer

- Module: M5.5
- Description of the module: Named entity recognizer, EntityPro: it discovers the named entities in a text and classifies them. The available categories are person (PER), organization (ORG), geopolical entity (GPE) and location (LOC). This module uses a machine learning approach but its output can be corrected using white/black lists. English models have been trained using CoNLL 2003 dataset (84.49 F1). Output annotation is represented in the IOB2 format. EntityPro is part of TextPro.
- Input layer(s): Tokens, POS-tags, lemma
- Output layer(s): named entities
- Required modules: Tokenizer, POS-tagger, Lemmatizer
- Level of operation: sentence level
- Language dependent: yes

3.6 Word Sense Disambiguation

The word sense disambiguation task can be performed by means of two different modules: The UKB based WSD and the SVM based WSD. The UKB based WSD is the integrated one in the prototype.

3.6.1 UKB based WSD

- Module: M6.1
- Description of the module: UKB is a collection of programs for performing graph-based Word Sense Disambiguation. UKB applies the so-called Personalized PageRank on a Lexical Knowledge Base (LKB) to rank the vertices of the LKB and thus perform disambiguation. UKB has been developed by the IXA group. The module accepts lemmatized and POS tagged text in NAF format as standard input and outputs NAF.

- Input layer(s): lemma, POS
- Output layer(s): synsets
- Required modules: POS-tagger module
- Level of operation: sentence
- Language dependent: yes

3.6.2 SVM based WSD

- Module: M6.2
- Description of the module: WSD-VUA is a machine learning system that performs Word Sense Disambiguation. It is based on a supervised machine learning approach: Suppor Vector Machines. The library selected for the low level machine learning engine is SVM-light[28]. A bag-of-words feature model is used for representing the training instances, where tokens and lemmas are considered to build the context for each target word. One classifier is build for each target lemma, following the one-vs-all approach to deal with the multilabel classification of whe WSD task, as SVM is in principle a binary classifier. The relative frequency of a feature with respect to a certain classifier is considered to filter out to general features, and also for weighting each feature in the training phase.
- Input layer(s): token and term (lemmas and POS)
- Output layer(s): token and term (extended with word senses information)
- Required modules: tokenizer, pos-tagger, lemmatizar
- Level of operation: sentence and document
- Language dependent: no

3.7 Named Entity Disambiguation

3.7.1 Spotlight based NED

- Module: M7.1
- Description of the module: This module performs the Named Entity Disambiguation task based on DBpedia Spotlight. Assuming that a DBpedia Spotlight Rest server for a given language is locally running, the module will take NAF as input (containing elements) and perform Named Entity Disambiguation for your language of choice.

[28] `http://svmlight.joachims.org/`

Developed by IXA NLP Group. For English the Wikipedia dump from March 7, 2012[29] is used. The module accepts text with named entities in NAF format as standard input, it disambiguates them and outputs them in NAF.

- Input layer(s): named entities
- Output layer(s): disambiguated named entities
- Required modules: NERC module
- Level of operation: sentence level
- Language dependent: yes

3.8 Coreference Resolution

3.8.1 Graph-based Coreference

- Module: M8.1
- Description of the module: The CorefGraph-en module provides an implementation of the Multi-Sieve Pass system for for Coreference Resolution system originally proposed by the Stanford NLP Group [Raghunathan *et al.*, 2010; Lee *et al.*, 2011] and [Lee *et al.*, 2013]. This system proposes a number of deterministic passes, ranging from high precision to higher recall, each dealing with a different manner in which coreference manifests itself in running text. The evaluation of the Multi Sieve Pass on CoNLL 2011 obtains: 50.1 (MUC), 66.5 (B^2), 50.6 ($CEAF_m$), 68.4 (BLANC) and 52.3 (CoNLL F1) values. The implementation is a result of a collaboration between the IXA NLP and LinguaMedia Groups.[30]. The module provides coreference resolution for various languages such as English and Spanish. The module accepts parsed text in NAF format as standard input and outputs the enriched text in NAF.
- Input layer(s): lemma, POS, named-entities, constituents
- Output layer(s): coreferences
- Required modules: tokenizer, POS-tagger and NERC modules
- Level of operation: document level
- Language dependent: yes

[29] `http://dumps.wikimedia.org/enwiki/20120307/`
[30] http://linguamedia.deusto.es

3.8.2 Toponym resolution system

- Module: M8.2
- Description of the module: This system implements a mapping between the geographical location identified by EntityPro and an unambiguous spatial footprint of the same place, e.g. its geographic latitude/longitude coordinates [Buscaldi and Magnini, 2010]. It uses a Word Sense Disambiguation approach based on Geonames[31] and GoogleMaps[32] information. The underlying algorithm is language independent. The coverage of the module on the news is 87%. The evaluation of the system accuracy gets 79.6% on 39 English Wikipedia pages (3363 toponyms).
- Input layer(s): Tokens, Named Entities
- Output layer(s): Toponym coordinates
- Required modules: Tokenizer, named entity recognizer
- Level of operation: geo-referenced locations
- Language dependent: no

3.9 Sematic Role Labeling

3.9.1 Mate-tools based SRL

- Module: M9.1
- Description of the module: This module is based on the MATE-tools [Björkelund *et al.*, 2010], a pipeline of linguistic processors that performs lemmatization, part-of-speech tagging, dependency parsing, and semantic role labeling of a sentence. They report on the CoNLL 2009 Shared Task a labelled semantic F1 of 85.63 for English and 79.91 for Spanish. Due to the input of the module is a NAF file that includes lemmatization, pos-tagging and dependency parsing, the module only implements the semantic role labeler [Björkelund *et al.*, 2009]. The module is ready to work with Spanish and English. The module accepts parsed text in NAF format as standard input and outputs the enriched text in NAF.
- Input layer(s): lemma, POS, dependencies
- Output layer(s): semantic roles
- Required modules: tokenizer, POS-tagger, dependency parsing modules
- Level of operation: sentence level
- Language dependent: yes

[31] `http://www.geonames.org/`
[32] `https://maps.google.it/`

3.10 Event Classification

- Module: M10.1
- Description of the module: This module use the OntoTagger and NAFKybot software specifically developed within the project. Both programs are re-implementations of software that was initially developed in the European KYOTO project (ICT-211423)[33] in Java1.6, including various new features. The OntoTagger program is used to insert a semantic typing of nouns, verbs and adjectives into the NAF-term layer. The semantic classification comes from the KYOTO-DOLCE ontology and from the predicate-matrix resource. Both resources have mappings to wordnets through which the appropriate semantic classifications are assigned to synsets in the NAF term layer. The NAFKybot program is used to run patterns over the NAF structure and to extract 3 types of predicates with their roles. The predicates belong to the classes Communication, Cognition and Other. The latter class represents the domain specific actions mentioned in the documents. Predicates are selected on the basis of the semantic types from the predicate-matrix and the syntactic dependencies. The roles are based on the subject-object relations to these predicates. Time and location entities are also added as roles. The roles are labeled using PropBank labels: a1 and a2, and additional as l1 for Places, l2 for Countries and t1 for time indicators. The approach is unsupervised and requires a term layer with synsets and a dependency layer in NAF. To run the event-classification the following resources are needed:
  - Ontotag tables that match values from the predicate-matrix to synset identifiers.
  - A set of profiles that combine predicate-matrix values with syntactic dependencies for subject and object relations

  These resources are available for English and can easily be derived for other languages. The software modules and the English resources are available on github:
  - OntoTagger: `https://github.com/cltl/OntoTagger.git`
  - NAFKybot: `https://github.com/cltl/KafKybot.git`

  A VirtualMachine with the combined module is being released soon.
- Input layer(s): Terms, Dependencies, Time and Location entities
- Output layer(s): Predicates and Roles
- Required modules: Dependency parser that usually also includes the creation of a term layer, Word-sense-disambiguation on a term layer, Named-entity-recognition for Time and Locations.

[33] `http://www.kyoto-project.eu`

- Level of operation: sentence level
- Language dependence: yes

3.11 Factuality

- Module: M11.1
- Description of the module: The factuality module classifies for each event whether it is factual or not. The module uses the MAchine Learning for LanguagE Toolkit Mallet [McCallum, 2002] (version 2.0.7) trained on FactBank v1.0[34] to determine the factuality of an event in text.
- Input layer(s): tokenized and POS-tagged text
- Output layer(s): factuality layer with token spans
- Required modules: Tokenization and POS tagging
- Level of operation: sentence level
- Language dependent: yes

3.12 Discourse

- Module: M12.1
- Description of the module: The discourse module separates text about the main issue in the news article from side issues or background information.
- Input layer(s): News Industry Text Format (NITF)[35]
- Output layer(s): Discourse layer with token spans
- Required modules: -
- Level of operation: document and paragraph level
- Language dependent: no

[34] http://www.ldc.upenn.edu/Catalog/catalogEntry.jsp?catalogId=LDC2009T23
[35] http://www.iptc.org/site/News_Exchange_Formats/NITF/

3.13 Opinions

- Module: M13.1
- Description of the module: This module implements the Deluxe version of the opinion mining using Machine Learning and the crfsuite toolkit (`http://www.chokkan.org/software/crfsuite/`). The module aims to detect opinions and extract three elements for each opinion: a) Opinion expression; b) Opinion holder; c) Opinion target. This module works for English and Dutch, and it has been trained using a small corpus annotated manually by 2 annotators at the VUA. The input of this module has to be a NAF file, preferably with text, term (with pos and polarity), entity and property layers, as they will be used to extract the features for the system. In case there are some layers missing in the input NAF, the module will still work, but some features will not be available and the performance can be punished. The output is the NAF file extended with the opinion layer.

  The system works in 2 different steps, it first detects the opinion entities (expression, target and holder), and then detects the relations that link these entities:

  - Expression - Target: what the expression is about
  - Expression - Holder: who is stating the expression

  For the first task, we followed a sequence labelling task, as the three elements are made up be a span of tokens, and use Conditional Random Fields for infering the model. At the same time we split this task into 2 different task: we first detect opinion expressions (using base features as tokens, words, part-of-speech and polarity values), and then we reuse these opinion expressions as features for extending a traditional set of features (tokens, words, entities, aspects) and train the model for detecting holders and targets. Second for the relation learning, we use Support Vector Machines for infering binary classifiers that given an expression and a holder (or target) are able to say if there is actually a link between them or not. Once obtained the opinion entities and their relations, the opinion triples can be made up.
- Input layer(s): token, term (terms, part-of-speech, polarity information at term level, entity, constituent and dependency
- Output layer(s): opinion layer
- Required modules:tokenizer, pos-tagger, lemmatizer, polarity tagger, parser
- Level of operation: sentence and document
- Language dependent: no

3.14 Pipelines for English

This section presents various English pipelines used to test the performance of different infrastructures for text processing. Section 3.14.1 presents the main pipeline which will be used in the prototype for event detection. Sections 3.14.2 and 3.14.3 show alternative pipelines developed and tested at VUA and FBK respectively. The pipelines are described in two different ways. First, the dependencies among each task (module) are presented. Then, the order of the linguistic processors is displayed.

3.14.1 IXA-pipe based Pipeline

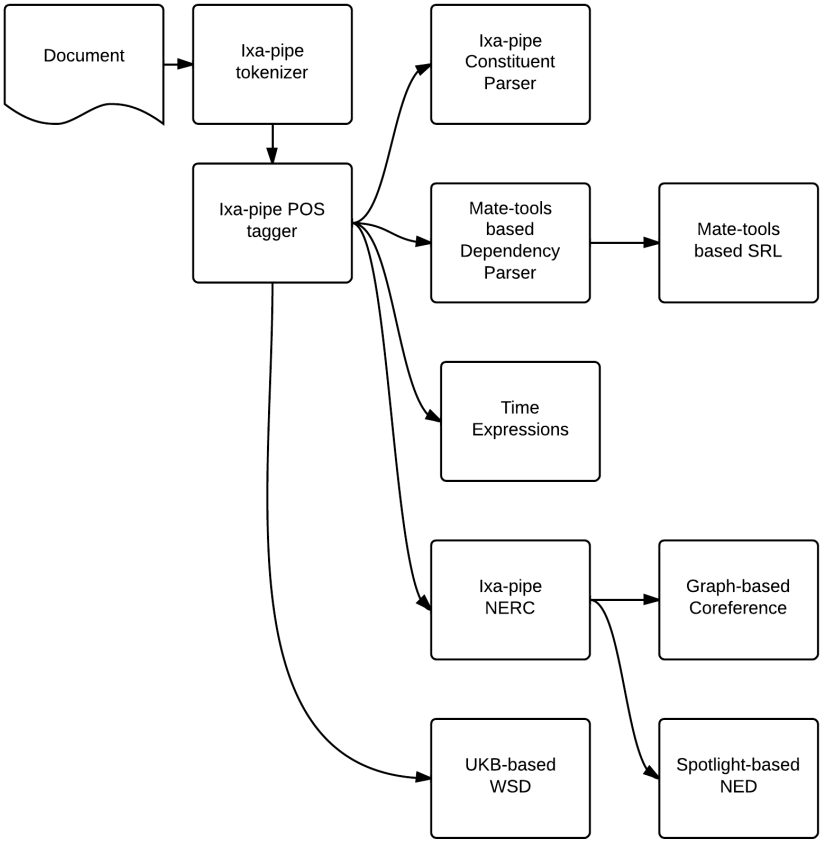

Figure 1: IXA-pipe based pipeline. Dependencies among modules

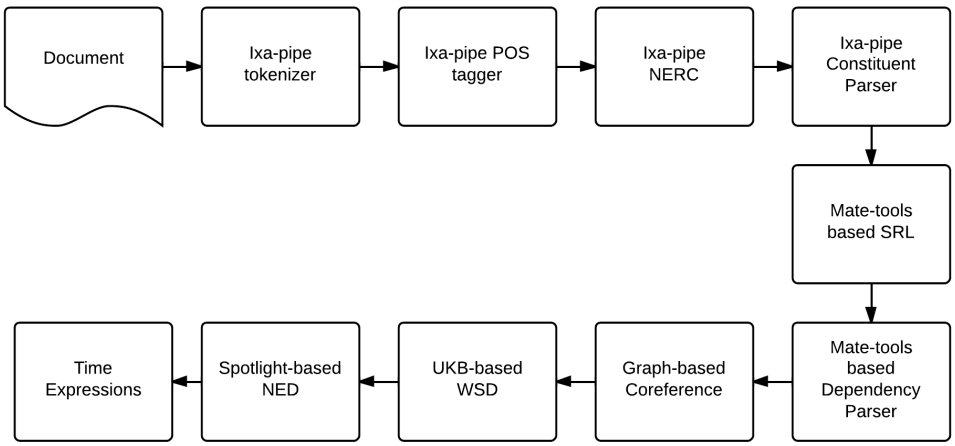

Figure 2: IXA-pipe based pipeline. Linguistic processors' order

3.14.2 Stanford based Pipeline

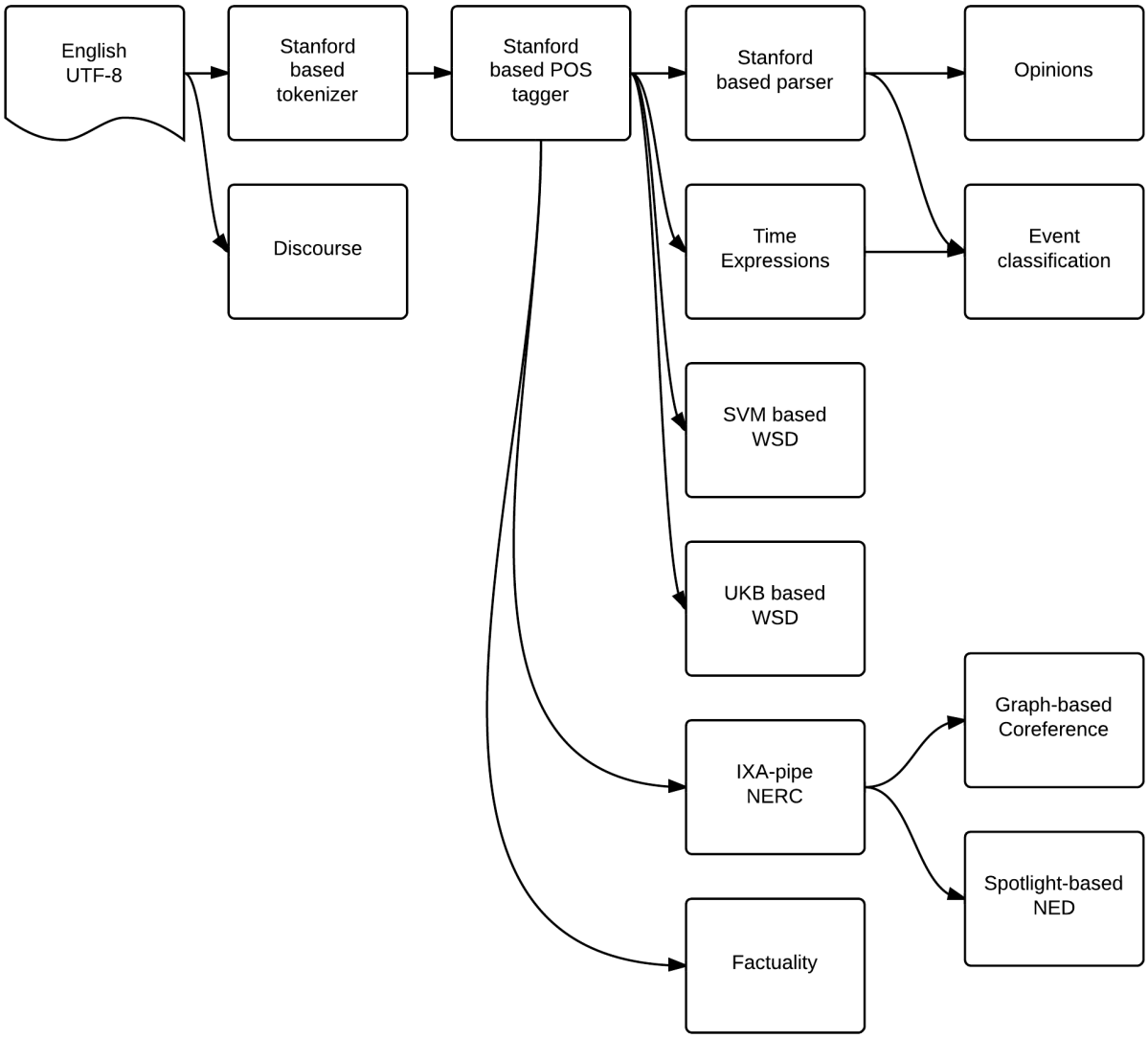

Figure 3: Stanford-based pipeline. Dependencies among modules

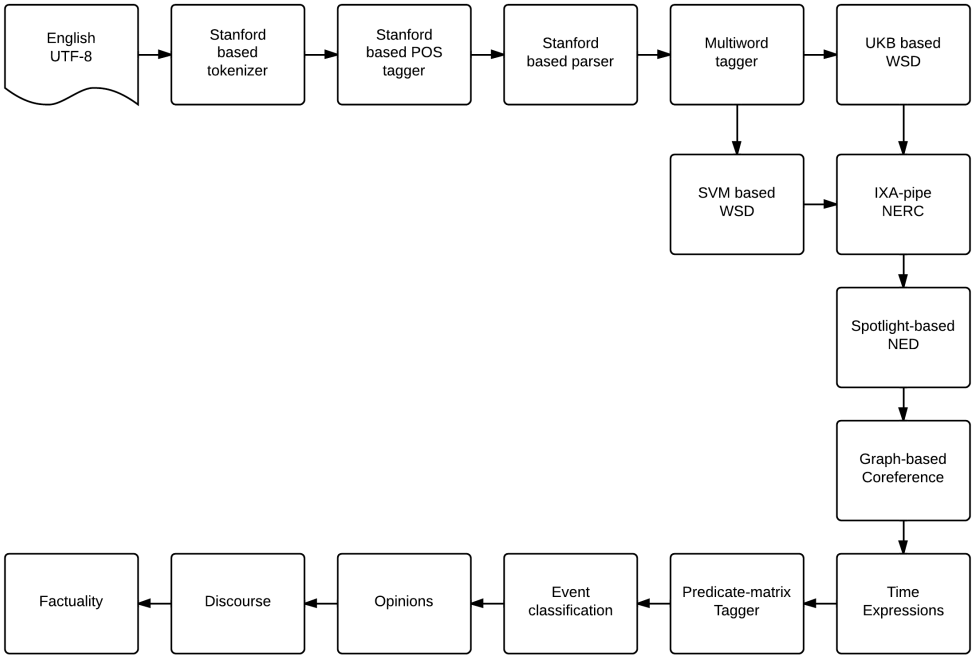

Figure 4: Stanford-based pipeline. Linguistic processors' order

3.14.3 TextPro based Pipeline

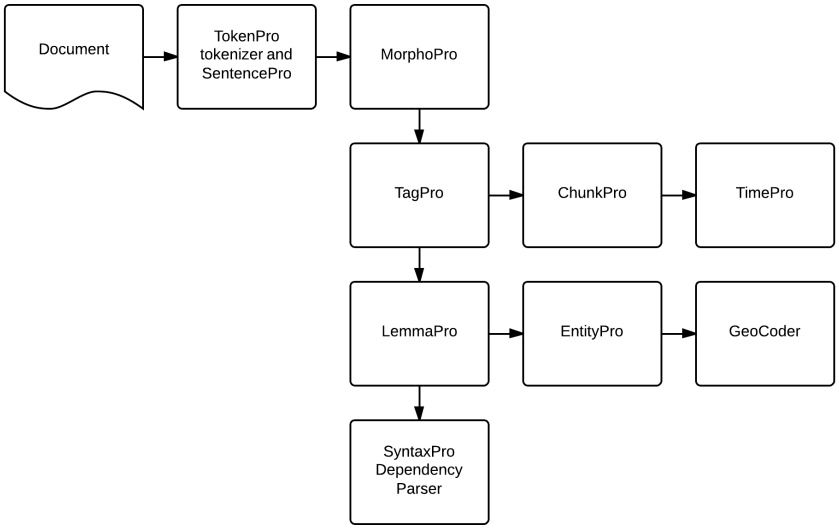

Figure 5: TextPro-based pipeline. Dependencies among modules

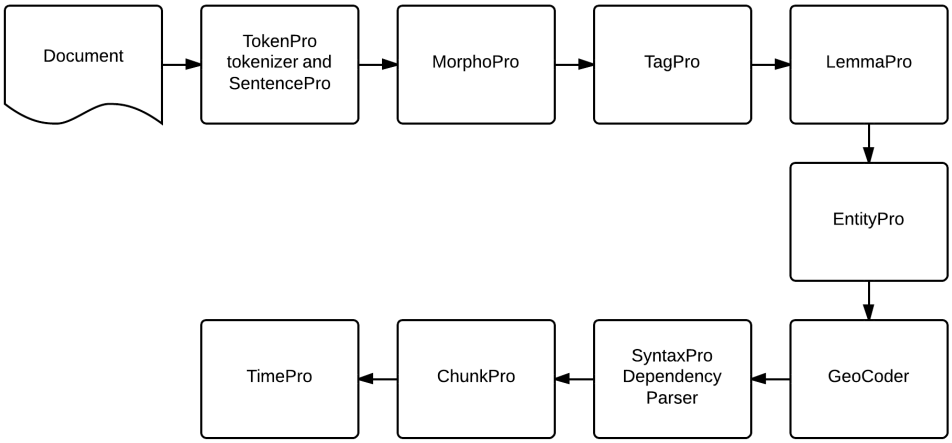

Figure 6: TextPro-based pipeline. Linguistic processors' order

4 Dutch NLP Processing

The NLP processing for Dutch is similar to the Stanford based Pipeline for English. Instead of the Stanford parser for English, we use the Alpino parser[36].

[36]http://www.let.rug.nl/vannoord/alp/Alpino/

1. Syntactic-depedency parser: UTF-8 text is processed by the Alpino-parser. The output is converted to NAF layers for tokens, terms, chunks, constituents and dependencies

2. Word-sense-disambiguation: we use the UKB with the Dutch Wordnet to assign synsets to the term layer.

3. Time and location detection: we use the KYOTO-named-entity library to detect time and locations.

4. Event-classification: the predicate-matrix for English is converted to a predicate-matrix for Dutch using the wordnet-equivalence relations. The English profiles for the NAFKybot have been converted to profiles that use the dependency labels from the Alpino output.

5. Factuality classification: whilst the module is set up in a generic manner, training data in languages other than English is still needed. In M9-12 the NWR benchmark datasets will also be marked up with factuality information, which will be used as input for the Dutch, Spanish and Italian versions of the factuality module.

5 Italian NLP Processing

The NLP processing suite for Italian is similar to the TextPro based pipeline for English. In the following lines, we briefly describe each module of the Italian pipeline:

1. Italian TokenPro: the module provides sentence segmentation and tokenization for Italian as explained in section 3.1.3.

2. Italian MorphoPro and TagPro: the morphological analyzer and POS tagger, described in the section 3.2.3, comes with two language models, Italian and English. MorphoPro has an Italian dictionary with 1,878,285 analyses for 149,372 lemmas. The Italian PoS-Tagger model is trained on a corpus using a subset of the ELRA tagset. It was the best system at EVALITA 2007 (98.04% F1).

3. Italian ChunkPro: this module performs chunking as explained in section 3.3.4. It assigns the Italian tokens to one of these 2 categories: NP (noun phrase) or VX (verb phrase).

4. Italian Named Entity recognizer: EntityPro performs name entity recognition as described in section 3.5.2. The Italian model is trained on ICAB [Magnini *et al.*, 2006] and it was the best performing at EVALITA 2008 (82.1% F1). The module for system training is included in the distribution, and also the customization through white/black lists is possible.

5. Italian TimePro: the Italian version of this module is under development. It will be built like the module described in section 3.4.

6. Italian SyntaxPro: this module implements an Italian Dependency Parser. It is based on Malt Parser [37], a language-independent system for data-driven dependency parsing written in Java (open source). It was evaluated at Evalita 2011, using the Turin University Treebank for training (88.62% LAS, 92.85% UAS).

6 Spanish NLP Processing

The NLP processing for Spanish will be similar to the IXA-pipe based pipeline for English. In the following lines, we briefly describe each module of the pipeline for Spanish:

1. Ixa-pipe tokenizer: the module provides sentence segmentation and tokenization for Spanish as explained in section 3.1.1.

2. IXA-pipe POS tagger: the module provides lemmatization and POS tagging for Spanish as explained in section 3.2.1. Spanish POS tagging models are Maximum Entropy models. The models have been trained and evaluated using the Ancora corpus; it was randomly divided in 90% for training (440K words) and 10% testing (70K words), obtaining a performance of 98.88%. The Spanish dictionary options for lemmatization are: a) Plain text dictionary: es-lemmas.dict; b) Morfologik stemming: spanish.dict.

3. IXA-pipe parser: the module provides constituent parsing for Spanish as explained in section 3.3.1. Spanish models have been trained using the Ancora corpus (`http://clic.ub.edu/corpus/en/ancora`).

4. Mate-tools based parser: the module provides dependency parsing for Spanish as explained in section 3.3.2.

5. Ixa-pipe NERC: the module provides named entities for Spanish as explained in section 3.5.1. Spanish models have been trained using CoNLL 2002 dataset (79.92 F1).

6. UKB based WSD: the module performs word sense disambiguation as explained in section 3.6.1. For Spanish we use the Spanish WordNed to perform the disambiguation.

7. Spotlight based NED: the module performs named entity disambiguation as explained in section 3.7.1. For Spanish the latest Spanish Wikipedia dump is used.

8. Graph based Coreference: the module performs coreference resolution as explained in section 3.8.1.

[37] `http://maltparser.org/`

9. TimePro based time parsing: the module presented in section 3.4 needs to be trained to obtain a Spanish time parsing module.

10. Mate-tools based SRL: the module provides sematic roles for Spanish as explained in section 3.9.1. The modules have been trained using the Ancora corpus.

11. Factuality classification: whilst the module is set up in a generic manner, training data in languages other than English is still needed. In M9-12 the NWR benchmark datasets will also be marked up with factuality information, which will be used as input for the Dutch, Spanish and Italian versions of the factuality module.

7 Scaling of Text Processing

There is a continuous increase of computational power that produces an overwhelming flow of data which calls for a paradigm shift in computing architecture and large scale data processing. Within the NewsReader project we foresee an estimating flow of 2 million News article per day, and the linguistic analysis of those documents needs to be done in a reasonable time frame (one or few hours). The project faces thus an important challenge regarding the scalability of the linguistic processing of texts.

In this section, we describe the initial experiments performed with the goal of analyzing the scaling capabilities of our NLP processing pipeline. We first describe the adoption of a framework for streaming computing, analyzing the requirements of most of our modules. We also report some initial results of the expected performance gain when running the LP processing using a stream computing framework.

7.1 Storm

Storm is a framework for streaming computing whose aim is to implement highly scalable and parallel processing of data streams. According to its homepage[38]:

> Storm is a free and open source distributed realtime computation system. Storm makes it easy to reliably process unbounded streams of data, doing for realtime processing what Hadoop did for batch processing. Storm is simple, can be used with any programming language, and is a lot of fun to use!
>
> Storm has many use cases: realtime analytics, online machine learning, continuous computation, distributed RPC, ETL, and more. Storm is fast: a benchmark clocked it at over a million tuples processed per second per node. It is scalable, fault-tolerant, guarantees your data will be processed, and is easy to set up and operate.

[38] `http://storm-project.net/`

Storm processing is based on *topologies*: a graph of computation which processes messages forever (unlike Hadoop jobs, which follow a batch processing paradigm and finish when the job is done). Storm topologies are thus top level abstractions which describe the processing that each message undergoes. Topology nodes fall into two categories: the so called *spout* and *bolt* nodes. *Spout* nodes are the entry points of topology and the source of the initial messages to be processed. *Bolt* nodes are the actual processing units, which receive messages, process them, and pass the processed messages to the next stage in the topology.

The data model of Storm is the *tuple*, i.e., each node in the topology consumes[39] and produces tuples. The tuple is an abstraction of the data model, and is general enough to allow any data to be passed around the topology.

In Storm, each node of the topology may reside on a different physical machine; the Storm controller (called *Nimbus*) is the responsible to send the tuples among the different machines, and guarantees that each message undergoes all the nodes in the topology.

It is important to note that Storm allows several instances of each topology node, thus allowing the actual parallel processing. Following the so called *parallelism hint*, it is possible to specify how many instances of each topology node will be actually running.

In our experiments, we use the following setting:

- The *spout* node is a process which reads a text document and sends it to the first *bolt* of the topology.
- the *bolt* nodes are wrapper programs which receive input tuples, call the actual NLP modules for processing, and send the output tuples to the next stage in the topology.
- The tuples in our Storm topology comprise two elements, a document identifier and the document itself, encoded as a string with the XML serialization of the NAF document.

7.2 Experiment setting

We created a small NLP pipeline comprising four modules: a tokenizer (tok, see Section 3.1.1, Ixa-pipe Tokenizer), a part of speech tagger (pos, see section 3.2.1, Ixa-pipe POS tagger), a Named Entity recognition module (ned, see Section 3.5.1, Ixa-pipe Named Entity Recognizer) and a word sense disambiguation module (wsd, see Section 3.6.1, UKB based WSD).

Initially the four modules are executed following a pipeline architecture, i.e., each module running sequentially one after the other. This setting is the baseline system and the starting point for our analysis.

On a second experiment, we implement a Storm topology following again a pipeline approach. This setting is similar to the baseline system but has a main advantage. When a module finishes the processing, it passes the annotated document to the next step, and

[39] Unlike *spout* nodes, which are the initial nodes and therefore do not consume tuples.

	Total time	words per sec	sent. per sec.	gain
Four documents				
pipeline	1m11.385s	94.9 w/s	3.8 sent/s	–
Storm	1m11.372s	94.9 w/s	3.8 sent/s	0.0%
$Storm_3$	0m38.274s	177.0 w/s	7.1 sent/s	46.4%
$Storm_4$	0m33.540s	201.9 w/s	8.1 sent/s	53.0%
$Storm_5$	0m34.472s	196.5 w/s	7.9 sent/s	51.7%
Eight documents				
pipeline	2m8.625s	93.9 w/s	3.6 sent/s	–
Storm	1m57.277s	103.0 w/s	3.9 sent/s	8.8%
$Storm_3$	0m58.492s	206.5w/s	7.9s/s	54.5%
$Storm_4$	0m50.415s	239.6w/s	9.2s/s	60.8%
$Storm_5$	0m54.544s	221.4w/s	8.5s/s	57.6%

Table 3: Performance of the NLP pipeline in different settings. pipeline is the basic pipeline used as baseline. Storm is the same pipeline but run as a Storm topology. $Storm_3$ is the Storm pipeline with 3 instances of the WSD module ($Storm_4$ has 4 instances and $Storm_5$ 5 instances, respectively).

start processing the next document. Therefore, in this setting there are as many documents processed in parallel as stages in the pipeline. Because we have a pipeline comprising 4 modules, the pipeline is able to process 4 documents at the same time. On a final setting, we experiment creating many instances of some selected *bolt* nodes, therefore allowing the parallel execution of them.

All the experiments were performed on a PC machine with a Intel Core i5-3570 3.4GHz processor with 4 cores and 4GB RAM, running linux.

7.3 Results

We tested the NLP pipeline with 8 documents, each one comprising about 1500 words and 60 sentences. We performed experiments for 4 documents (6,773 words and 271 sentences) and the complete set of 8 documents (12,077 words and 462 sentences).

Table 3 shows the time elapsed by processing the documents. The first five rows correspond to the processing of 4 documents and the last five rows to the processing of 8 documents.

As the Table shows, the baseline system runs at a performance of about 95 words per second. The simple Storm topology yields little gain in performance (no gain at all with 4 documents, and a modest 8.8% with 8 documents). This little gain can be explained by the following reasons:

- Storm spends a considerable amount of time loading the topology and initializing the modules. As the results show, this initial overhead is big when processing a small number of documents, but it gets proportionally smaller as the number of document increases.

- There is an unbalance regarding the time spent of each module. For the 70 seconds needed to process the documents, almost 60 seconds were spent in the WSD module, which is by far the module needing more time to complete its task. Although the Storm topology can in principle multiply the performance by a factor of four (the number of modules ran in parallel), in practice all the computing is concentrated in one single node, which severely compromises the overall performance gain.

With these points in mind, we experimented three alternatives (dubbed $Storm_3$, $Storm_4$ and $Storm_5$), with 3 instances (respectively, 4 and 5 instances) of the WSD module running in parallel. The results in Table 3 show that running multiple instances of WSD does increase the overall performance significatively. The major gain is obtained with four copies of WSD, with an increase of 60% in the overall performance. This result is expected, giving the fact that the machine where the experiments were ran has 4 CPU cores.

All in all, this initial experiments have shown that there is a big room for improvement regarding NLP processing performance when adopting parallel architectures such as Storm. With the use of large clusters with many nodes, we expect a significant boost in the performance of overall NLP processing.

In future experiments we want to try the following settings:

- Non linear topologies. The experiments described here follow the pipeline approach, but in principle we could also consider executing non linear topologies, where two modules are processing the same document at the same time. Non linear topologies require considering the following aspects:
  - We need to clearly identify the pre- and post-requisites of each module, thus deducting the indications as to which modules must precede which and which modules can be ran in parallel on the same document.
  - We need a special *bolt* which receives input from many NLP modules *bolts* (each one conveying different annotations on the same document) and *merges* all this information producing a single, unified document.
- Granularity. NLP modules work at different type of granularity. For instance, a POS tagger works at a sentence level, the WSD module works at paragraph level, whereas a coreference module works at a document level. We want to experiment splitting the input document into pieces of the required granularity, so that the NLP modules can quickly analyze those pieces, thus increasing the overall processing speed.

8 Conclusions

This deliverable describes the first version of the **Event Detection** framework developed in NewsReader to process large and continuous streams of English, Dutch, Spanish and Italian news articles.

First, we focused on our current findings to allow the interoperability of predicate models. During the first cycle of the NewsReader project (Event Detection, version 1) we

focused on processing general English news. The process of detecting events is divided into various tasks. Each task is executed by one module and various modules are available to perform each task:

- Tokenization: Ixa-pipe tokenizer, Stanford-based tokenizer, TokenPro-based tokenizer
- POS-tagging: Ixa-pipe POS tagger, Stanford-based POS tagger, TextPro-based POS tagger
- Parsing: Ixa-pipe parser, Mate-tools based parser, Stanford-based parser, ChunkPro-based parser
- Time expressions: TimePro-based module
- Named Entity Recognition: Ixa-pipe Named Entity Recognizer, EntityPro-based Named Entity Recognizer
- Word Sense Disambiguation: UKB based WSD, SVM based WSD
- Named Entity Disambiguation: Spotlight based NED
- Coreference Resolution: Graph-based Coreference, Toponym resolution system
- Sematic Role Labeling: Mate-tools based SRL
- Event classification: event classification module
- Factuality: factuality module
- Discourse: discourse module
- Opinions: opinion module

We considered three different advanced English pipelines: IXA-pipeline [40], TextPro pipeline based on FBK TextPro[41] and CoreNLP pipeline based on Stanford CoreNLP[42].

Among the three available pipelines, we decided to start extending the IXA-pipeline for developing the English pipeline because it offers an open source, data-centric, modular, robust and efficient set of NLP tools for Spanish and English. It can be used "as is" or exploit its modularity to pick and change different components or add new ones. IXA pipeline currently provides the following linguistic annotations: Sentence segmentation, tokenization (Ixa-pipe tokenizer), Part of Speech (POS) tagging (Ixa-pipe POS tagger), Lemmatization, Named Entity Recognition and Classification (Ixa-pipe NER), Syntactic Parsing (Ixa-pipe parser) and Coreference Resolution (Graph-based coreference). To this

[40] `https://github.com/ixa-ehu`
[41] `http://textpro.fbk.eu/`
[42] `http://nlp.stanford.edu/software/corenlp.shtml`

basic pipeline we also have added new modules for Semantic Role Labelling (Mate-tools based SRL), Named Entity Disambiguation (Spotlight based NED), TimeML annotations (TimePro-based module), factuality (factuality module), event classificatio, discourse and opinion mining. This is the first prototype to process events in English. We are going to perform an exhaustive evaluation of each module in order to set the best option for English in terms of performance and accuracy. Thus, it is possible that some of the modules that includes the current pipeline will be replaced based on the evaluation results.

We also described the initial Spanish, Italian and Dutch processing pipelines.

Finally, we also presented our initial plan for testing the performance of different scaling infrastructures for advanced NLP processing. We already integrated the whole English pipleline into a KVM virtual machine in order to facilitate its deployment in clusters or the cloud. Given its open-source nature, it can also be modified and extended for it to work with other languages.

A English pipeline - Output example

Given the input text:

> *Autos sales within the eurozone have, on average, declined by 1.2% within the same period, according to the Warsaw Business Journal – demonstrating weaknesses in the export model followed by Poland in times when their customer base is suffering serious economic decline and uncertainty.*

The pipelines for English produce various layers in which different linguistic information is represented in NAF format. Following, the output of the IXA-pipe based pipeline (section 3.14.1) is presented.

```
<?xml version="1.0" encoding="UTF-8"?>
<NAF xml:lang="en" version="v1.opener">
  <nafHeader>
    <linguisticProcessors layer="text">
      <lp name="ixa-pipe-tok-en" timestamp="2013-10-28 16:38:06" version="1.0" />
    </linguisticProcessors>
    <linguisticProcessors layer="terms">
      <lp name="ixa-pipe-pos-en" timestamp="2013-10-28 16:38:21" version="1.0" />
    </linguisticProcessors>
    <linguisticProcessors layer="terms">
      <lp name="ukb" version="2.0.22.g0edc5ba" timestamp="2013-10-28T16:43:35Z"/>
    </linguisticProcessors>
    <linguisticProcessors layer="entities">
      <lp name="ixa-pipe-nerc-en" timestamp="2013-10-28 16:42:45" version="1.0" />
    </linguisticProcessors>
    <linguisticProcessors layer="constituency">
      <lp name="ixa-pipe-parse-en" timestamp="2013-10-28 16:40:00" version="1.0" />
    </linguisticProcessors>
    <linguisticProcessors layer="deps">
      <lp name="ixa-pipe-srl-en" timestamp="2013-10-28 16:49:51" version="1.0" />
    </linguisticProcessors>
    <linguisticProcessors layer="srl">
      <lp name="ixa-pipe-srl-en" timestamp="2013-10-28 16:49:51" version="1.0" />
    </linguisticProcessors>
    <linguisticProcessors layer="ned">
      <lp name="ixa-pipe-spotlight" timestamp="2013-10-28 16:42:57" version="1.0" />
    </linguisticProcessors>
    <linguisticProcessors layer="timex3">
      <lp name="TimePro" timestamp="2013-10-28 16:44:25.97" version="2.0"/>
    </linguisticProcessors>
  </nafHeader>
  <text>
    <wf id="w1" sent="1" offset="0" length="5">Autos</wf>
    <wf id="w2" sent="1" offset="6" length="5">sales</wf>
    <wf id="w3" sent="1" offset="12" length="6">within</wf>
    <wf id="w4" sent="1" offset="19" length="3">the</wf>
    <wf id="w5" sent="1" offset="23" length="8">eurozone</wf>
    <wf id="w6" sent="1" offset="32" length="4">have</wf>
    <wf id="w7" sent="1" offset="36" length="1">,</wf>
    <wf id="w8" sent="1" offset="38" length="2">on</wf>
    <wf id="w9" sent="1" offset="41" length="7">average</wf>
    <wf id="w10" sent="1" offset="48" length="1">,</wf>
    <wf id="w11" sent="1" offset="50" length="8">declined</wf>
    <wf id="w12" sent="1" offset="59" length="2">by</wf>
    <wf id="w13" sent="1" offset="62" length="3">1.2</wf>
```

```
    <wf id="w14" sent="1" offset="65" length="1">%</wf>
    <wf id="w15" sent="1" offset="67" length="6">within</wf>
    <wf id="w16" sent="1" offset="74" length="3">the</wf>
    <wf id="w17" sent="1" offset="78" length="4">same</wf>
    <wf id="w18" sent="1" offset="83" length="6">period</wf>
    <wf id="w19" sent="1" offset="89" length="1">,</wf>
    <wf id="w20" sent="1" offset="91" length="9">according</wf>
    <wf id="w21" sent="1" offset="101" length="2">to</wf>
    <wf id="w22" sent="1" offset="104" length="3">the</wf>
    <wf id="w23" sent="1" offset="108" length="6">Warsaw</wf>
    <wf id="w24" sent="1" offset="115" length="8">Business</wf>
    <wf id="w25" sent="1" offset="124" length="7">Journal</wf>
    <wf id="w26" sent="1" offset="132" length="2">--</wf>
    <wf id="w27" sent="1" offset="135" length="13">demonstrating</wf>
    <wf id="w28" sent="1" offset="149" length="10">weaknesses</wf>
    <wf id="w29" sent="1" offset="160" length="2">in</wf>
    <wf id="w30" sent="1" offset="163" length="3">the</wf>
    <wf id="w31" sent="1" offset="167" length="6">export</wf>
    <wf id="w32" sent="1" offset="174" length="5">model</wf>
    <wf id="w33" sent="1" offset="180" length="8">followed</wf>
    <wf id="w34" sent="1" offset="189" length="2">by</wf>
    <wf id="w35" sent="1" offset="192" length="6">Poland</wf>
    <wf id="w36" sent="1" offset="199" length="2">in</wf>
    <wf id="w37" sent="1" offset="202" length="5">times</wf>
    <wf id="w38" sent="1" offset="208" length="4">when</wf>
    <wf id="w39" sent="1" offset="213" length="5">their</wf>
    <wf id="w40" sent="1" offset="219" length="8">customer</wf>
    <wf id="w41" sent="1" offset="228" length="4">base</wf>
    <wf id="w42" sent="1" offset="233" length="2">is</wf>
    <wf id="w43" sent="1" offset="236" length="9">suffering</wf>
    <wf id="w44" sent="1" offset="246" length="7">serious</wf>
    <wf id="w45" sent="1" offset="254" length="8">economic</wf>
    <wf id="w46" sent="1" offset="263" length="7">decline</wf>
    <wf id="w47" sent="1" offset="271" length="3">and</wf>
    <wf id="w48" sent="1" offset="275" length="11">uncertainty</wf>
    <wf id="w49" sent="1" offset="286" length="1">.</wf>
  </text>
  <terms>
    <!--Autos-->
    <term id="t1" type="open" lemma="auto" pos="N" morphofeat="NNS">
      <span>
        <target id="w1"/>
      </span>
    <externalReferences>
    <externalRef resource="wn30g.bin64" reference="eng-v30-02958343-n" confidence="1"/>
    </externalReferences></term>
    <!--sales-->
    <term id="t2" type="open" lemma="sale" pos="N" morphofeat="NNS">
      <span>
        <target id="w2"/>
      </span>
    <externalReferences>
    <externalRef resource="wn30g.bin64" reference="eng-v30-14564306-n" confidence="↩
        0.208793"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-01117541-n" confidence="↩
        0.207858"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-01114824-n" confidence="↩
        0.207129"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-01117723-n" confidence="↩
        0.192372"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-06527851-n" confidence="↩
        0.183849"/>
    </externalReferences>
    </term>
    <!--within-->
```

```
<term id="t3" type="close" lemma="within" pos="P" morphofeat="IN">
  <span>
    <target id="w3"/>
  </span>
</term>
<!--the-->
<term id="t4" type="close" lemma="the" pos="D" morphofeat="DT">
  <span>
    <target id="w4"/>
  </span>
</term>
<!--eurozone-->
<term id="t5" type="open" lemma="eurozone" pos="N" morphofeat="NN">
  <span>
    <target id="w5"/>
  </span>
</term>
<!--have-->
<term id="t6" type="open" lemma="have" pos="V" morphofeat="VBP">
  <span>
    <target id="w6"/>
  </span>
<externalReferences>
<externalRef resource="wn30g.bin64" reference="eng-v30-00056930-v" confidence="↩
    0.105518"/>
<externalRef resource="wn30g.bin64" reference="eng-v30-02203362-v" confidence="↩
    0.0907798"/>
<externalRef resource="wn30g.bin64" reference="eng-v30-02204692-v" confidence="↩
    0.081659"/>
<externalRef resource="wn30g.bin64" reference="eng-v30-00770437-v" confidence="↩
    0.0778677"/>
<externalRef resource="wn30g.bin64" reference="eng-v30-02236124-v" confidence="↩
    0.0665086"/>
<externalRef resource="wn30g.bin64" reference="eng-v30-01733477-v" confidence="↩
    0.0626347"/>
<externalRef resource="wn30g.bin64" reference="eng-v30-02630189-v" confidence="↩
    0.0614596"/>
<externalRef resource="wn30g.bin64" reference="eng-v30-01156834-v" confidence="↩
    0.0592568"/>
<externalRef resource="wn30g.bin64" reference="eng-v30-00065639-v" confidence="↩
    0.0499097"/>
<externalRef resource="wn30g.bin64" reference="eng-v30-02355596-v" confidence="↩
    0.0482167"/>
<externalRef resource="wn30g.bin64" reference="eng-v30-02210119-v" confidence="↩
    0.0472105"/>
<externalRef resource="wn30g.bin64" reference="eng-v30-02108026-v" confidence="↩
    0.0439075"/>
<externalRef resource="wn30g.bin64" reference="eng-v30-00121046-v" confidence="↩
    0.0378816"/>
<externalRef resource="wn30g.bin64" reference="eng-v30-02378453-v" confidence="↩
    0.0358956"/>
<externalRef resource="wn30g.bin64" reference="eng-v30-00065370-v" confidence="↩
    0.0291182"/>
<externalRef resource="wn30g.bin64" reference="eng-v30-02205098-v" confidence="↩
    0.0280949"/>
<externalRef resource="wn30g.bin64" reference="eng-v30-00120796-v" confidence="↩
    0.0274307"/>
<externalRef resource="wn30g.bin64" reference="eng-v30-01427127-v" confidence="↩
    0.0237571"/>
<externalRef resource="wn30g.bin64" reference="eng-v30-02740745-v" confidence="↩
    0.0228931"/>
</externalReferences></term>
<!--,-->
<term id="t7" type="close" lemma="," pos="O" morphofeat=",">
  <span>
```

```
      <target id="w7"/>
    </span>
  </term>
  <!--on-->
  <term id="t8" type="close" lemma="on" pos="P" morphofeat="IN">
    <span>
      <target id="w8"/>
    </span>
  </term>
  <!--average-->
  <term id="t9" type="open" lemma="average" pos="N" morphofeat="NN">
    <span>
      <target id="w9"/>
    </span>
  <externalReferences>
  <externalRef resource="wn30g.bin64" reference="eng-v30-05856979-n" confidence="↩
      0.398104"/>
  <externalRef resource="wn30g.bin64" reference="eng-v30-06021761-n" confidence="↩
      0.326869"/>
  <externalRef resource="wn30g.bin64" reference="eng-v30-13820655-n" confidence="↩
      0.275027"/>
  </externalReferences></term>
  <!--,-->
  <term id="t10" type="close" lemma="," pos="O" morphofeat=",">
    <span>
      <target id="w10"/>
    </span>
  </term>
  <!--declined-->
  <term id="t11" type="open" lemma="decline" pos="V" morphofeat="VBD">
    <span>
      <target id="w11"/>
    </span>
  <externalReferences>
  <externalRef resource="wn30g.bin64" reference="eng-v30-00431826-v" confidence="↩
      0.235793"/>
  <externalRef resource="wn30g.bin64" reference="eng-v30-02237338-v" confidence="↩
      0.168388"/>
  <externalRef resource="wn30g.bin64" reference="eng-v30-00203866-v" confidence="↩
      0.146447"/>
  <externalRef resource="wn30g.bin64" reference="eng-v30-00982913-v" confidence="↩
      0.120833"/>
  <externalRef resource="wn30g.bin64" reference="eng-v30-02039876-v" confidence="↩
      0.119456"/>
  <externalRef resource="wn30g.bin64" reference="eng-v30-00797430-v" confidence="↩
      0.118107"/>
  <externalRef resource="wn30g.bin64" reference="eng-v30-01971603-v" confidence="↩
      0.0909747"/>
  </externalReferences>
  </term>
  <!--by-->
  <term id="t12" type="close" lemma="by" pos="P" morphofeat="IN">
    <span>
      <target id="w12"/>
    </span>
  </term>
  <!--1.2-->
  <term id="t13" type="close" lemma="1.2" pos="O" morphofeat="CD">
    <span>
      <target id="w13"/>
    </span>
  </term>
  <!--%-->
  <term id="t14" type="open" lemma="%" pos="N" morphofeat="NN">
    <span>
```

```
      <target id="w14"/>
    </span>
  </term>
  <!--within-->
  <term id="t15" type="close" lemma="within" pos="P" morphofeat="IN">
    <span>
      <target id="w15"/>
    </span>
  </term>
  <!--the-->
  <term id="t16" type="close" lemma="the" pos="D" morphofeat="DT">
    <span>
      <target id="w16"/>
    </span>
  </term>
  <!--same-->
  <term id="t17" type="open" lemma="same" pos="G" morphofeat="JJ">
    <span>
      <target id="w17"/>
    </span>
  <externalReferences>
  <externalRef resource="wn30g.bin64" reference="eng-v30-02068476-a" confidence="↩
      0.297714"/>
  <externalRef resource="wn30g.bin64" reference="eng-v30-02062670-a" confidence="↩
      0.297614"/>
  <externalRef resource="wn30g.bin64" reference="eng-v30-01411065-a" confidence="↩
      0.238183"/>
  <externalRef resource="wn30g.bin64" reference="eng-v30-00355611-a" confidence="↩
      0.166489"/>
  </externalReferences>
  </term>
  <!--period-->
  <term id="t18" type="open" lemma="period" pos="N" morphofeat="NN">
    <span>
      <target id="w18"/>
    </span>
  <externalReferences>
  <externalRef resource="wn30g.bin64" reference="eng-v30-15113229-n" confidence="↩
      0.335494"/>
  <externalRef resource="wn30g.bin64" reference="eng-v30-15247518-n" confidence="↩
      0.126179"/>
  <externalRef resource="wn30g.bin64" reference="eng-v30-15144178-n" confidence="↩
      0.112635"/>
  <externalRef resource="wn30g.bin64" reference="eng-v30-13513747-n" confidence="↩
      0.110827"/>
  <externalRef resource="wn30g.bin64" reference="eng-v30-15258281-n" confidence="↩
      0.108465"/>
  <externalRef resource="wn30g.bin64" reference="eng-v30-06843520-n" confidence="↩
      0.10555"/>
  <externalRef resource="wn30g.bin64" reference="eng-v30-15289779-n" confidence="↩
      0.100849"/>
  </externalReferences>
  </term>
  <!--,-->
  <term id="t19" type="close" lemma="," pos="O" morphofeat=",">
    <span>
      <target id="w19"/>
    </span>
  </term>
  <!--according-->
  <term id="t20" type="open" lemma="accord" pos="V" morphofeat="VBG">
    <span>
      <target id="w20"/>
    </span>
  <externalReferences>
```

```
<externalRef resource="wn30g.bin64" reference="eng-v30-02255268-v" confidence="↩
    0.506088"/>
<externalRef resource="wn30g.bin64" reference="eng-v30-02700104-v" confidence="↩
    0.493912"/>
</externalReferences>
</term>
<!--to-->
<term id="t21" type="close" lemma="to" pos="P" morphofeat="TO">
  <span>
    <target id="w21"/>
  </span>
</term>
<!--the-->
<term id="t22" type="close" lemma="the" pos="D" morphofeat="DT">
  <span>
    <target id="w22"/>
  </span>
</term>
<!--Warsaw-->
<term id="t23" type="close" lemma="Warsaw" pos="R" morphofeat="NNP">
  <span>
    <target id="w23"/>
  </span>
</term>
<!--Business-->
<term id="t24" type="close" lemma="Business" pos="R" morphofeat="NNP">
  <span>
    <target id="w24"/>
  </span>
</term>
<!--Journal-->
<term id="t25" type="close" lemma="Journal" pos="R" morphofeat="NNP">
  <span>
    <target id="w25"/>
  </span>
</term>
<!--- - -->
<term id="t26" type="close" lemma="--" pos="O" morphofeat=":">
  <span>
    <target id="w26"/>
  </span>
</term>
<!--demonstrating-->
<term id="t27" type="open" lemma="demonstrating" pos="N" morphofeat="NN">
  <span>
    <target id="w27"/>
  </span>
</term>
<!--weaknesses-->
<term id="t28" type="open" lemma="weakness" pos="N" morphofeat="NNS">
  <span>
    <target id="w28"/>
  </span>
<externalReferences>
<externalRef resource="wn30g.bin64" reference="eng-v30-05040275-n" confidence="↩
    0.219284"/>
<externalRef resource="wn30g.bin64" reference="eng-v30-05204982-n" confidence="↩
    0.212166"/>
<externalRef resource="wn30g.bin64" reference="eng-v30-14462946-n" confidence="↩
    0.208531"/>
<externalRef resource="wn30g.bin64" reference="eng-v30-14474718-n" confidence="↩
    0.186656"/>
<externalRef resource="wn30g.bin64" reference="eng-v30-07498614-n" confidence="↩
    0.173362"/>
</externalReferences>
```

```
    </term>
    <!--in-->
    <term id="t29" type="close" lemma="in" pos="P" morphofeat="IN">
      <span>
        <target id="w29"/>
      </span>
    </term>
    <!--the-->
    <term id="t30" type="close" lemma="the" pos="D" morphofeat="DT">
      <span>
        <target id="w30"/>
      </span>
    </term>
    <!--export-->
    <term id="t31" type="open" lemma="export" pos="N" morphofeat="NN">
      <span>
        <target id="w31"/>
      </span>
    <externalReferences>
    <externalRef resource="wn30g.bin64" reference="eng-v30-03306207-n" confidence="1"/>
    </externalReferences>
    </term>
    <!--model-->
    <term id="t32" type="open" lemma="model" pos="N" morphofeat="NN">
      <span>
        <target id="w32"/>
      </span>
    <externalReferences>
    <externalRef resource="wn30g.bin64" reference="eng-v30-05925366-n" confidence="↩
        0.131938"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-03777283-n" confidence="↩
        0.128121"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-10324851-n" confidence="↩
        0.124701"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-05937112-n" confidence="↩
        0.116263"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-10324560-n" confidence="↩
        0.11153"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-05890249-n" confidence="↩
        0.109472"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-10291240-n" confidence="↩
        0.100462"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-00898804-n" confidence="↩
        0.0932642"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-05845652-n" confidence="↩
        0.0842488"/>
    </externalReferences>
    </term>
    <!--followed-->
    <term id="t33" type="open" lemma="follow" pos="V" morphofeat="VBN">
      <span>
        <target id="w33"/>
      </span>
    <externalReferences>
    <externalRef resource="wn30g.bin64" reference="eng-v30-02445925-v" confidence="↩
        0.0659959"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-02455407-v" confidence="↩
        0.0634114"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-02542280-v" confidence="↩
        0.0608754"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-02000868-v" confidence="↩
        0.0581441"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-02720354-v" confidence="↩
        0.0553292"/>
```

```
<externalRef resource="wn30g.bin64" reference="eng-v30-02406585-v" confidence="↩
    0.0548257"/>
<externalRef resource="wn30g.bin64" reference="eng-v30-02720544-v" confidence="↩
    0.053056"/>
<externalRef resource="wn30g.bin64" reference="eng-v30-00150776-v" confidence="↩
    0.0502942"/>
<externalRef resource="wn30g.bin64" reference="eng-v30-00729109-v" confidence="↩
    0.0456989"/>
<externalRef resource="wn30g.bin64" reference="eng-v30-01744450-v" confidence="↩
    0.044826"/>
<externalRef resource="wn30g.bin64" reference="eng-v30-02346895-v" confidence="↩
    0.0446978"/>
<externalRef resource="wn30g.bin64" reference="eng-v30-01998432-v" confidence="↩
    0.0407605"/>
<externalRef resource="wn30g.bin64" reference="eng-v30-02712772-v" confidence="↩
    0.0386861"/>
<externalRef resource="wn30g.bin64" reference="eng-v30-02600255-v" confidence="↩
    0.0379948"/>
<externalRef resource="wn30g.bin64" reference="eng-v30-02720697-v" confidence="↩
    0.0341946"/>
<externalRef resource="wn30g.bin64" reference="eng-v30-02720149-v" confidence="↩
    0.0320276"/>
<externalRef resource="wn30g.bin64" reference="eng-v30-01728355-v" confidence="↩
    0.0318747"/>
<externalRef resource="wn30g.bin64" reference="eng-v30-02625339-v" confidence="↩
    0.0315772"/>
<externalRef resource="wn30g.bin64" reference="eng-v30-00118764-v" confidence="↩
    0.0299774"/>
<externalRef resource="wn30g.bin64" reference="eng-v30-00589738-v" confidence="↩
    0.0286651"/>
<externalRef resource="wn30g.bin64" reference="eng-v30-02198602-v" confidence="↩
    0.0255733"/>
<externalRef resource="wn30g.bin64" reference="eng-v30-00351406-v" confidence="↩
    0.0245801"/>
<externalRef resource="wn30g.bin64" reference="eng-v30-02561697-v" confidence="↩
    0.0237631"/>
<externalRef resource="wn30g.bin64" reference="eng-v30-01991931-v" confidence="↩
    0.0231707"/>
</externalReferences>
</term>
<!--by-->
<term id="t34" type="close" lemma="by" pos="P" morphofeat="IN">
  <span>
    <target id="w34"/>
  </span>
</term>
<!--Poland-->
<term id="t35" type="close" lemma="Poland" pos="R" morphofeat="NNP">
  <span>
    <target id="w35"/>
  </span>
</term>
<!--in-->
<term id="t36" type="close" lemma="in" pos="P" morphofeat="IN">
  <span>
    <target id="w36"/>
  </span>
</term>
<!--times-->
<term id="t37" type="open" lemma="time" pos="N" morphofeat="NNS">
  <span>
    <target id="w37"/>
  </span>
<externalReferences>
```

```
    <externalRef resource="wn30g.bin64" reference="eng-v30-00028270-n" confidence="↩
        0.171904"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-15122231-n" confidence="↩
        0.126584"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-07309599-n" confidence="↩
        0.10711"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-15224692-n" confidence="↩
        0.100126"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-15129927-n" confidence="↩
        0.0967133"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-15270431-n" confidence="↩
        0.0935462"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-15135822-n" confidence="↩
        0.0875867"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-04991738-n" confidence="↩
        0.0872045"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-15245515-n" confidence="↩
        0.0774379"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-07288215-n" confidence="↩
        0.0517876"/>
    </externalReferences>
    </term>
    <!--when-->
    <term id="t38" type="close" lemma="when" pos="O" morphofeat="WRB">
      <span>
        <target id="w38"/>
      </span>
    </term>
    <!--their-->
    <term id="t39" type="close" lemma="their" pos="Q" morphofeat="PRP$">
      <span>
        <target id="w39"/>
      </span>
    </term>
    <!--customer-->
    <term id="t40" type="open" lemma="customer" pos="N" morphofeat="NN">
      <span>
        <target id="w40"/>
      </span>
    <externalReferences>
    <externalRef resource="wn30g.bin64" reference="eng-v30-09984659-n" confidence="1"/>
    </externalReferences>
    </term>
    <!--base-->
    <term id="t41" type="open" lemma="base" pos="N" morphofeat="NN">
      <span>
        <target id="w41"/>
      </span>
    <externalReferences>
    <externalRef resource="wn30g.bin64" reference="eng-v30-08013845-n" confidence="↩
        0.0850396"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-03387016-n" confidence="↩
        0.0723938"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-05793554-n" confidence="↩
        0.0677248"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-03569964-n" confidence="↩
        0.0616323"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-14618253-n" confidence="↩
        0.0602443"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-08490402-n" confidence="↩
        0.059251"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-13809769-n" confidence="↩
        0.0562555"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-02798290-n" confidence="↩
        0.0547097"/>
```

```
    <externalRef resource="wn30g.bin64" reference="eng-v30-06300193-n" confidence="↩
        0.0543966"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-02797881-n" confidence="↩
        0.0537931"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-14964590-n" confidence="↩
        0.0515831"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-08511777-n" confidence="↩
        0.0491601"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-02798117-n" confidence="↩
        0.0414417"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-02797692-n" confidence="↩
        0.0385336"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-06658118-n" confidence="↩
        0.0348133"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-02798574-n" confidence="↩
        0.0344422"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-13597794-n" confidence="↩
        0.0339005"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-02798769-n" confidence="↩
        0.0323842"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-09215315-n" confidence="↩
        0.0307365"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-13897837-n" confidence="↩
        0.0275642"/>
    </externalReferences>
    </term>
    <!--is-->
    <term id="t42" type="open" lemma="be" pos="V" morphofeat="VBZ">
      <span>
        <target id="w42"/>
      </span>
    <externalReferences>
    <externalRef resource="wn30g.bin64" reference="eng-v30-02604760-v" confidence="↩
        0.148452"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-02620587-v" confidence="↩
        0.131242"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-02655135-v" confidence="↩
        0.0988915"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-02445925-v" confidence="↩
        0.0861568"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-02664769-v" confidence="↩
        0.0844465"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-02603699-v" confidence="↩
        0.079442"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-02697725-v" confidence="↩
        0.0682691"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-02614181-v" confidence="↩
        0.0598959"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-02702508-v" confidence="↩
        0.0591133"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-02744820-v" confidence="↩
        0.0509246"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-02268246-v" confidence="↩
        0.0482347"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-02749904-v" confidence="↩
        0.0455293"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-02616386-v" confidence="↩
        0.0394022"/>
    </externalReferences>
    </term>
    <!--suffering-->
    <term id="t43" type="open" lemma="suffer" pos="V" morphofeat="VBG">
      <span>
        <target id="w43"/>
      </span>
```

```
    <externalReferences>
    <externalRef resource="wn30g.bin64" reference="eng-v30-00668099-v" confidence="↩
        0.16416"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-00065070-v" confidence="↩
        0.109943"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-00065639-v" confidence="↩
        0.10779"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-02109190-v" confidence="↩
        0.0971637"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-02121511-v" confidence="↩
        0.0969964"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-01794668-v" confidence="↩
        0.0882747"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-00204750-v" confidence="↩
        0.0800503"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-00078401-v" confidence="↩
        0.0755447"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-00204872-v" confidence="↩
        0.0713023"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-02110082-v" confidence="↩
        0.0597648"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-02719807-v" confidence="↩
        0.0490095"/>
    </externalReferences>
    </term>
    <!--serious-->
    <term id="t44" type="open" lemma="serious" pos="G" morphofeat="JJ">
      <span>
        <target id="w44"/>
      </span>
    <externalReferences>
    <externalRef resource="wn30g.bin64" reference="eng-v30-02118379-a" confidence="↩
        0.257781"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-02123314-a" confidence="↩
        0.164414"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-00651039-a" confidence="↩
        0.150677"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-01333477-a" confidence="↩
        0.149521"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-00748359-a" confidence="↩
        0.142432"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-01279611-a" confidence="↩
        0.135176"/>
    </externalReferences>
    </term>
    <!--economic-->
    <term id="t45" type="open" lemma="economic" pos="G" morphofeat="JJ">
      <span>
        <target id="w45"/>
      </span>
    <externalReferences>
    <externalRef resource="wn30g.bin64" reference="eng-v30-00840212-a" confidence="↩
        0.217377"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-01871565-a" confidence="↩
        0.21253"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-02716739-a" confidence="↩
        0.202872"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-02577454-a" confidence="↩
        0.198098"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-02716605-a" confidence="↩
        0.169124"/>
    </externalReferences>
    </term>
    <!--decline-->
    <term id="t46" type="open" lemma="decline" pos="N" morphofeat="NN">
```

```
      <span>
        <target id="w46"/>
      </span>
    <externalReferences>
    <externalRef resource="wn30g.bin64" reference="eng-v30-14422488-n" confidence="↩
        0.277992"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-13457378-n" confidence="↩
        0.270528"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-09265620-n" confidence="↩
        0.242071"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-13456567-n" confidence="↩
        0.20941"/>
    </externalReferences>
    </term>
    <!--and-->
    <term id="t47" type="close" lemma="and" pos="C" morphofeat="CC">
      <span>
        <target id="w47"/>
      </span>
    </term>
    <!--uncertainty-->
    <term id="t48" type="open" lemma="uncertainty" pos="N" morphofeat="NN">
      <span>
        <target id="w48"/>
      </span>
    <externalReferences>
    <externalRef resource="wn30g.bin64" reference="eng-v30-05698247-n" confidence="↩
        0.547585"/>
    <externalRef resource="wn30g.bin64" reference="eng-v30-04756887-n" confidence="↩
        0.452415"/>
    </externalReferences>
    </term>
    <!--.-->
    <term id="t49" type="close" lemma="." pos="O" morphofeat=".">
      <span>
        <target id="w49"/>
      </span>
    </term>
  </terms>
  <entities>
    <entity id="e1" type="organization">
      <references>
        <!--Warsaw Business Journal-->
        <span>
          <target id="t23" />
          <target id="t24" />
          <target id="t25" />
        </span>
      </references>
      <externalReferences>
        <externalRef resource="spotlight_v1" reference="http://dbpedia.org/resource/↩
            Warsaw_Business_Journal" />
      </externalReferences>
    </entity>
    <entity id="e2" type="location">
      <references>
        <!--Poland-->
        <span>
          <target id="t35" />
        </span>
      </references>
      <externalReferences>
        <externalRef resource="spotlight_v1" reference="http://dbpedia.org/resource/↩
            Poland_%28novel%29" />
      </externalReferences>
```

```
    </entity>
  </entities>
  <constituency>
    <tree>
      <!--Non-terminals-->
      <nt id="nter1" label="TOP" />
      <nt id="nter2" label="S" />
      <nt id="nter3" label="NP" />
      <nt id="nter4" label="NP" />
      <nt id="nter5" label="NNS" />
      <nt id="nter6" label="NNS" />
      <nt id="nter7" label="PP" />
      <nt id="nter8" label="IN" />
      <nt id="nter9" label="NP" />
      <nt id="nter10" label="DT" />
      <nt id="nter11" label="NN" />
      <nt id="nter12" label="VP" />
      <nt id="nter13" label="VBP" />
      <nt id="nter14" label="," />
      <nt id="nter15" label="PP" />
      <nt id="nter16" label="IN" />
      <nt id="nter17" label="NP" />
      <nt id="nter18" label="NN" />
      <nt id="nter19" label="," />
      <nt id="nter20" label="VP" />
      <nt id="nter21" label="VBD" />
      <nt id="nter22" label="PP" />
      <nt id="nter23" label="IN" />
      <nt id="nter24" label="NP" />
      <nt id="nter25" label="CD" />
      <nt id="nter26" label="NN" />
      <nt id="nter27" label="PP" />
      <nt id="nter28" label="IN" />
      <nt id="nter29" label="NP" />
      <nt id="nter30" label="DT" />
      <nt id="nter31" label="JJ" />
      <nt id="nter32" label="NN" />
      <nt id="nter33" label="," />
      <nt id="nter34" label="PP" />
      <nt id="nter35" label="VBG" />
      <nt id="nter36" label="PP" />
      <nt id="nter37" label="TO" />
      <nt id="nter38" label="NP" />
      <nt id="nter39" label="DT" />
      <nt id="nter40" label="NNP" />
      <nt id="nter41" label="NNP" />
      <nt id="nter42" label="NNP" />
      <nt id="nter43" label=":" />
      <nt id="nter44" label="S" />
      <nt id="nter45" label="VP" />
      <nt id="nter46" label="VBG" />
      <nt id="nter47" label="NP" />
      <nt id="nter48" label="NP" />
      <nt id="nter49" label="NP" />
      <nt id="nter50" label="NNS" />
      <nt id="nter51" label="PP" />
      <nt id="nter52" label="IN" />
      <nt id="nter53" label="NP" />
      <nt id="nter54" label="DT" />
      <nt id="nter55" label="NN" />
      <nt id="nter56" label="NN" />
      <nt id="nter57" label="VP" />
      <nt id="nter58" label="VBD" />
      <nt id="nter59" label="PP" />
      <nt id="nter60" label="IN" />
```

```
<nt id="nter61" label="NP" />
<nt id="nter62" label="NNP" />
<nt id="nter63" label="PP" />
<nt id="nter64" label="IN" />
<nt id="nter65" label="NP" />
<nt id="nter66" label="NNS" />
<nt id="nter67" label="SBAR" />
<nt id="nter68" label="WHADVP" />
<nt id="nter69" label="WRB" />
<nt id="nter70" label="S" />
<nt id="nter71" label="NP" />
<nt id="nter72" label="PRP$" />
<nt id="nter73" label="NN" />
<nt id="nter74" label="NN" />
<nt id="nter75" label="VP" />
<nt id="nter76" label="VBZ" />
<nt id="nter77" label="VP" />
<nt id="nter78" label="VBG" />
<nt id="nter79" label="NP" />
<nt id="nter80" label="NP" />
<nt id="nter81" label="JJ" />
<nt id="nter82" label="JJ" />
<nt id="nter83" label="NN" />
<nt id="nter84" label="CC" />
<nt id="nter85" label="NP" />
<nt id="nter86" label="NN" />
<nt id="nter87" label="." />
<!--Terminals-->
<!--Autos-->
<t id="ter1">
  <span>
    <target id="t1" />
  </span>
</t>
<!--sales-->
<t id="ter2">
  <span>
    <target id="t2" />
  </span>
</t>
<!--within-->
<t id="ter3">
  <span>
    <target id="t3" />
  </span>
</t>
<!--the-->
<t id="ter4">
  <span>
    <target id="t4" />
  </span>
</t>
<!--eurozone-->
<t id="ter5">
  <span>
    <target id="t5" />
  </span>
</t>
<!--have-->
<t id="ter6">
  <span>
    <target id="t6" />
  </span>
</t>
<!--,-->
```

```
<t id="ter7">
  <span>
    <target id="t7" />
  </span>
</t>
<!--on-->
<t id="ter8">
  <span>
    <target id="t8" />
  </span>
</t>
<!--average-->
<t id="ter9">
  <span>
    <target id="t9" />
  </span>
</t>
<!--,-->
<t id="ter10">
  <span>
    <target id="t10" />
  </span>
</t>
<!--declined-->
<t id="ter11">
  <span>
    <target id="t11" />
  </span>
</t>
<!--by-->
<t id="ter12">
  <span>
    <target id="t12" />
  </span>
</t>
<!--1.2-->
<t id="ter13">
  <span>
    <target id="t13" />
  </span>
</t>
<!--%-->
<t id="ter14">
  <span>
    <target id="t14" />
  </span>
</t>
<!--within-->
<t id="ter15">
  <span>
    <target id="t15" />
  </span>
</t>
<!--the-->
<t id="ter16">
  <span>
    <target id="t16" />
  </span>
</t>
<!--same-->
<t id="ter17">
  <span>
    <target id="t17" />
  </span>
</t>
```

```
<!--period-->
<t id="ter18">
  <span>
    <target id="t18" />
  </span>
</t>
<!--,-->
<t id="ter19">
  <span>
    <target id="t19" />
  </span>
</t>
<!--according-->
<t id="ter20">
  <span>
    <target id="t20" />
  </span>
</t>
<!--to-->
<t id="ter21">
  <span>
    <target id="t21" />
  </span>
</t>
<!--the-->
<t id="ter22">
  <span>
    <target id="t22" />
  </span>
</t>
<!--Warsaw-->
<t id="ter23">
  <span>
    <target id="t23" />
  </span>
</t>
<!--Business-->
<t id="ter24">
  <span>
    <target id="t24" />
  </span>
</t>
<!--Journal-->
<t id="ter25">
  <span>
    <target id="t25" />
  </span>
</t>
<!--  -  -->
<t id="ter26">
  <span>
    <target id="t26" />
  </span>
</t>
<!--demonstrating-->
<t id="ter27">
  <span>
    <target id="t27" />
  </span>
</t>
<!--weaknesses-->
<t id="ter28">
  <span>
    <target id="t28" />
  </span>
```

```
</t>
<!--in-->
<t id="ter29">
  <span>
    <target id="t29" />
  </span>
</t>
<!--the-->
<t id="ter30">
  <span>
    <target id="t30" />
  </span>
</t>
<!--export-->
<t id="ter31">
  <span>
    <target id="t31" />
  </span>
</t>
<!--model-->
<t id="ter32">
  <span>
    <target id="t32" />
  </span>
</t>
<!--followed-->
<t id="ter33">
  <span>
    <target id="t33" />
  </span>
</t>
<!--by-->
<t id="ter34">
  <span>
    <target id="t34" />
  </span>
</t>
<!--Poland-->
<t id="ter35">
  <span>
    <target id="t35" />
  </span>
</t>
<!--in-->
<t id="ter36">
  <span>
    <target id="t36" />
  </span>
</t>
<!--times-->
<t id="ter37">
  <span>
    <target id="t37" />
  </span>
</t>
<!--when-->
<t id="ter38">
  <span>
    <target id="t38" />
  </span>
</t>
<!--their-->
<t id="ter39">
  <span>
    <target id="t39" />
```

```
    </span>
  </t>
  <!--customer-->
  <t id="ter40">
    <span>
      <target id="t40" />
    </span>
  </t>
  <!--base-->
  <t id="ter41">
    <span>
      <target id="t41" />
    </span>
  </t>
  <!--is-->
  <t id="ter42">
    <span>
      <target id="t42" />
    </span>
  </t>
  <!--suffering-->
  <t id="ter43">
    <span>
      <target id="t43" />
    </span>
  </t>
  <!--serious-->
  <t id="ter44">
    <span>
      <target id="t44" />
    </span>
  </t>
  <!--economic-->
  <t id="ter45">
    <span>
      <target id="t45" />
    </span>
  </t>
  <!--decline-->
  <t id="ter46">
    <span>
      <target id="t46" />
    </span>
  </t>
  <!--and-->
  <t id="ter47">
    <span>
      <target id="t47" />
    </span>
  </t>
  <!--uncertainty-->
  <t id="ter48">
    <span>
      <target id="t48" />
    </span>
  </t>
  <!--.-->
  <t id="ter49">
    <span>
      <target id="t49" />
    </span>
  </t>
  <!--Tree edges-->
  <edge from="nter2" to="nter1" />
  <edge from="nter3" to="nter2" />
```

```
<edge from="nter4" to="nter3" />
<edge from="nter5" to="nter4" />
<edge from="ter1" to="nter5" />
<edge from="nter6" to="nter4" />
<edge from="ter2" to="nter6" />
<edge from="nter7" to="nter3" />
<edge from="nter8" to="nter7" />
<edge from="ter3" to="nter8" />
<edge from="nter9" to="nter7" />
<edge from="nter10" to="nter9" />
<edge from="ter4" to="nter10" />
<edge from="nter11" to="nter9" />
<edge from="ter5" to="nter11" />
<edge from="nter12" to="nter2" />
<edge from="nter13" to="nter12" />
<edge from="ter6" to="nter13" />
<edge from="nter14" to="nter12" />
<edge from="ter7" to="nter14" />
<edge from="nter15" to="nter12" />
<edge from="nter16" to="nter15" />
<edge from="ter8" to="nter16" />
<edge from="nter17" to="nter15" />
<edge from="nter18" to="nter17" />
<edge from="ter9" to="nter18" />
<edge from="nter19" to="nter12" />
<edge from="ter10" to="nter19" />
<edge from="nter20" to="nter12" />
<edge from="nter21" to="nter20" />
<edge from="ter11" to="nter21" />
<edge from="nter22" to="nter20" />
<edge from="nter23" to="nter22" />
<edge from="ter12" to="nter23" />
<edge from="nter24" to="nter22" />
<edge from="nter25" to="nter24" />
<edge from="ter13" to="nter25" />
<edge from="nter26" to="nter24" />
<edge from="ter14" to="nter26" />
<edge from="nter27" to="nter20" />
<edge from="nter28" to="nter27" />
<edge from="ter15" to="nter28" />
<edge from="nter29" to="nter27" />
<edge from="nter30" to="nter29" />
<edge from="ter16" to="nter30" />
<edge from="nter31" to="nter29" />
<edge from="ter17" to="nter31" />
<edge from="nter32" to="nter29" />
<edge from="ter18" to="nter32" />
<edge from="nter33" to="nter20" />
<edge from="ter19" to="nter33" />
<edge from="nter34" to="nter20" />
<edge from="nter35" to="nter34" />
<edge from="ter20" to="nter35" />
<edge from="nter36" to="nter34" />
<edge from="nter37" to="nter36" />
<edge from="ter21" to="nter37" />
<edge from="nter38" to="nter36" />
<edge from="nter39" to="nter38" />
<edge from="ter22" to="nter39" />
<edge from="nter40" to="nter38" />
<edge from="ter23" to="nter40" />
<edge from="nter41" to="nter38" />
<edge from="ter24" to="nter41" />
<edge from="nter42" to="nter38" />
<edge from="ter25" to="nter42" />
<edge from="nter43" to="nter20" />
```

```
<edge from="ter26" to="nter43" />
<edge from="nter44" to="nter20" />
<edge from="nter45" to="nter44" />
<edge from="nter46" to="nter45" />
<edge from="ter27" to="nter46" />
<edge from="nter47" to="nter45" />
<edge from="nter48" to="nter47" />
<edge from="nter49" to="nter48" />
<edge from="nter50" to="nter49" />
<edge from="ter28" to="nter50" />
<edge from="nter51" to="nter48" />
<edge from="nter52" to="nter51" />
<edge from="ter29" to="nter52" />
<edge from="nter53" to="nter51" />
<edge from="nter54" to="nter53" />
<edge from="ter30" to="nter54" />
<edge from="nter55" to="nter53" />
<edge from="ter31" to="nter55" />
<edge from="nter56" to="nter53" />
<edge from="ter32" to="nter56" />
<edge from="nter57" to="nter47" />
<edge from="nter58" to="nter57" />
<edge from="ter33" to="nter58" />
<edge from="nter59" to="nter57" />
<edge from="nter60" to="nter59" />
<edge from="ter34" to="nter60" />
<edge from="nter61" to="nter59" />
<edge from="nter62" to="nter61" />
<edge from="ter35" to="nter62" />
<edge from="nter63" to="nter57" />
<edge from="nter64" to="nter63" />
<edge from="ter36" to="nter64" />
<edge from="nter65" to="nter63" />
<edge from="nter66" to="nter65" />
<edge from="ter37" to="nter66" />
<edge from="nter67" to="nter57" />
<edge from="nter68" to="nter67" />
<edge from="nter69" to="nter68" />
<edge from="ter38" to="nter69" />
<edge from="nter70" to="nter67" />
<edge from="nter71" to="nter70" />
<edge from="nter72" to="nter71" />
<edge from="ter39" to="nter72" />
<edge from="nter73" to="nter71" />
<edge from="ter40" to="nter73" />
<edge from="nter74" to="nter71" />
<edge from="ter41" to="nter74" />
<edge from="nter75" to="nter70" />
<edge from="nter76" to="nter75" />
<edge from="ter42" to="nter76" />
<edge from="nter77" to="nter75" />
<edge from="nter78" to="nter77" />
<edge from="ter43" to="nter78" />
<edge from="nter79" to="nter77" />
<edge from="nter80" to="nter79" />
<edge from="nter81" to="nter80" />
<edge from="ter44" to="nter81" />
<edge from="nter82" to="nter80" />
<edge from="ter45" to="nter82" />
<edge from="nter83" to="nter80" />
<edge from="ter46" to="nter83" />
<edge from="nter84" to="nter79" />
<edge from="ter47" to="nter84" />
<edge from="nter85" to="nter79" />
<edge from="nter86" to="nter85" />
```

```
      <edge from="ter48" to="nter86" />
      <edge from="nter87" to="nter2" />
      <edge from="ter49" to="nter87" />
    </tree>
  </constituency>
  <srl>
    <!--t2 sales : A1[t1 Autos] AM-LOC[t3 within]-->
    <predicate id="pr1">
      <!--sales-->
      <span>
        <target id="t2" />
      </span>
      <externalReferences>
        <externalRef resource="NomBank" reference="sale.01" />
      </externalReferences>
      <role id="pr1r1" semRole="A1">
        <!--Autos-->
        <span>
          <target id="t1" head="yes" />
        </span>
      </role>
      <role id="pr1r2" semRole="AM-LOC">
        <!--within the eurozone-->
        <span>
          <target id="t3" head="yes" />
          <target id="t4" />
          <target id="t5" />
        </span>
      </role>
    </predicate>
    <!--t9 average : A1[t1 Autos] A2[t9 average]-->
    <predicate id="pr2">
      <!--average-->
      <span>
        <target id="t9" />
      </span>
      <externalReferences>
        <externalRef resource="NomBank" reference="average.01" />
      </externalReferences>
      <role id="pr2r1" semRole="A1">
        <!--Autos sales within the eurozone have-->
        <span>
          <target id="t1" />
          <target id="t2" head="yes" />
          <target id="t3" />
          <target id="t4" />
          <target id="t5" />
          <target id="t6" />
        </span>
      </role>
      <role id="pr2r2" semRole="A2">
        <!--average-->
        <span>
          <target id="t9" head="yes" />
        </span>
      </role>
    </predicate>
    <!--t11 declined : A1[t1 Autos] AM-ADV[t8 on] A2[t12 by] AM-TMP[t15 within] AM-ADV[↩
        t20 according]-->
    <predicate id="pr3">
      <!--declined-->
      <span>
        <target id="t11" />
      </span>
      <externalReferences>
```

```
      <externalRef resource="PropBank" reference="decline.01" />
    </externalReferences>
    <role id="pr3r1" semRole="A1">
      <!--Autos sales within the eurozone have-->
      <span>
        <target id="t1" />
        <target id="t2" head="yes" />
        <target id="t3" />
        <target id="t4" />
        <target id="t5" />
        <target id="t6" />
      </span>
    </role>
    <role id="pr3r2" semRole="AM-ADV">
      <!--on average-->
      <span>
        <target id="t8" head="yes" />
        <target id="t9" />
      </span>
    </role>
    <role id="pr3r3" semRole="A2">
      <!--by 1.2 %-->
      <span>
        <target id="t12" head="yes" />
        <target id="t13" />
        <target id="t14" />
      </span>
    </role>
    <role id="pr3r4" semRole="AM-TMP">
      <!--within the same period-->
      <span>
        <target id="t15" head="yes" />
        <target id="t16" />
        <target id="t17" />
        <target id="t18" />
      </span>
    </role>
    <role id="pr3r5" semRole="AM-ADV">
      <!--according to the Warsaw Business Journal -- demonstrating weaknesses in ↩
          the export model followed by Poland in times when their customer base is ↩
          suffering serious economic decline and uncertainty-->
      <span>
        <target id="t20" head="yes" />
        <target id="t21" />
        <target id="t22" />
        <target id="t23" />
        <target id="t24" />
        <target id="t25" />
        <target id="t26" />
        <target id="t27" />
        <target id="t28" />
        <target id="t29" />
        <target id="t30" />
        <target id="t31" />
        <target id="t32" />
        <target id="t33" />
        <target id="t34" />
        <target id="t35" />
        <target id="t36" />
        <target id="t37" />
        <target id="t38" />
        <target id="t39" />
        <target id="t40" />
        <target id="t41" />
        <target id="t42" />
```

```
        <target id="t43" />
        <target id="t44" />
        <target id="t45" />
        <target id="t46" />
        <target id="t47" />
        <target id="t48" />
      </span>
    </role>
  </predicate>
  <!--t28 weaknesses : A1[t22 the] A1[t27 demonstrating] A2[t22 the] A1[t29 in]-->
  <predicate id="pr4">
    <!--weaknesses-->
    <span>
      <target id="t28" />
    </span>
    <externalReferences>
      <externalRef resource="NomBank" reference="weakness.01" />
    </externalReferences>
    <role id="pr4r1" semRole="A1">
      <!--the Warsaw Business Journal-->
      <span>
        <target id="t22" />
        <target id="t23" />
        <target id="t24" />
        <target id="t25" head="yes" />
      </span>
    </role>
    <role id="pr4r2" semRole="A1">
      <!--demonstrating-->
      <span>
        <target id="t27" head="yes" />
      </span>
    </role>
    <role id="pr4r3" semRole="A2">
      <!--the Warsaw Business Journal -- demonstrating weaknesses in the export ↩
          model followed by Poland in times when their customer base is suffering ↩
          serious economic decline and uncertainty-->
      <span>
        <target id="t22" />
        <target id="t23" />
        <target id="t24" />
        <target id="t25" />
        <target id="t26" />
        <target id="t27" />
        <target id="t28" head="yes" />
        <target id="t29" />
        <target id="t30" />
        <target id="t31" />
        <target id="t32" />
        <target id="t33" />
        <target id="t34" />
        <target id="t35" />
        <target id="t36" />
        <target id="t37" />
        <target id="t38" />
        <target id="t39" />
        <target id="t40" />
        <target id="t41" />
        <target id="t42" />
        <target id="t43" />
        <target id="t44" />
        <target id="t45" />
        <target id="t46" />
        <target id="t47" />
        <target id="t48" />
```

```
      </span>
    </role>
    <role id="pr4r4" semRole="A1">
      <!--in the export model followed by Poland in times when their customer base is ↩
          suffering serious economic decline and uncertainty-->
      <span>
        <target id="t29" head="yes" />
        <target id="t30" />
        <target id="t31" />
        <target id="t32" />
        <target id="t33" />
        <target id="t34" />
        <target id="t35" />
        <target id="t36" />
        <target id="t37" />
        <target id="t38" />
        <target id="t39" />
        <target id="t40" />
        <target id="t41" />
        <target id="t42" />
        <target id="t43" />
        <target id="t44" />
        <target id="t45" />
        <target id="t46" />
        <target id="t47" />
        <target id="t48" />
      </span>
    </role>
  </predicate>
  <!--t32 model : A1[t31 export]-->
  <predicate id="pr5">
    <!--model-->
    <span>
      <target id="t32" />
    </span>
    <externalReferences>
      <externalRef resource="NomBank" reference="model.02" />
    </externalReferences>
    <role id="pr5r1" semRole="A1">
      <!--export-->
      <span>
        <target id="t31" head="yes" />
      </span>
    </role>
  </predicate>
  <!--t33 followed : A1[t30 the] A0[t34 by] AM-TMP[t36 in]-->
  <predicate id="pr6">
    <!--followed-->
    <span>
      <target id="t33" />
    </span>
    <externalReferences>
      <externalRef resource="PropBank" reference="follow.01" />
    </externalReferences>
    <role id="pr6r1" semRole="A1">
      <!--the export model followed by Poland in times when their customer base is ↩
          suffering serious economic decline and uncertainty-->
      <span>
        <target id="t30" />
        <target id="t31" />
        <target id="t32" head="yes" />
        <target id="t33" />
        <target id="t34" />
        <target id="t35" />
        <target id="t36" />
```

```
        <target id="t37" />
        <target id="t38" />
        <target id="t39" />
        <target id="t40" />
        <target id="t41" />
        <target id="t42" />
        <target id="t43" />
        <target id="t44" />
        <target id="t45" />
        <target id="t46" />
        <target id="t47" />
        <target id="t48" />
      </span>
    </role>
    <role id="pr6r2" semRole="A0">
      <!--by Poland-->
      <span>
        <target id="t34" head="yes" />
        <target id="t35" />
      </span>
    </role>
    <role id="pr6r3" semRole="AM-TMP">
      <!--in times when their customer base is suffering serious economic decline and ↩
          uncertainty-->
      <span>
        <target id="t36" head="yes" />
        <target id="t37" />
        <target id="t38" />
        <target id="t39" />
        <target id="t40" />
        <target id="t41" />
        <target id="t42" />
        <target id="t43" />
        <target id="t44" />
        <target id="t45" />
        <target id="t46" />
        <target id="t47" />
        <target id="t48" />
      </span>
    </role>
  </predicate>
  <!--t41 base : A1[t39 their] A0[t40 customer]-->
  <predicate id="pr7">
    <!--base-->
    <span>
      <target id="t41" />
    </span>
    <externalReferences>
      <externalRef resource="NomBank" reference="base.06" />
    </externalReferences>
    <role id="pr7r1" semRole="A1">
      <!--their-->
      <span>
        <target id="t39" head="yes" />
      </span>
    </role>
    <role id="pr7r2" semRole="A0">
      <!--customer-->
      <span>
        <target id="t40" head="yes" />
      </span>
    </role>
  </predicate>
  <!--t43 suffering : A0[t39 their] A1[t44 serious]-->
  <predicate id="pr8">
```

```
      <!--suffering-->
      <span>
        <target id="t43" />
      </span>
      <externalReferences>
        <externalRef resource="PropBank" reference="suffer.01" />
      </externalReferences>
      <role id="pr8r1" semRole="A0">
        <!--their customer base-->
        <span>
          <target id="t39" />
          <target id="t40" />
          <target id="t41" head="yes" />
        </span>
      </role>
      <role id="pr8r2" semRole="A1">
        <!--serious economic decline and uncertainty-->
        <span>
          <target id="t44" />
          <target id="t45" />
          <target id="t46" head="yes" />
          <target id="t47" />
          <target id="t48" />
        </span>
      </role>
    </predicate>
    <!--t46 decline : AM-MNR[t44 serious] A1[t45 economic]-->
    <predicate id="pr9">
      <!--decline-->
      <span>
        <target id="t46" />
      </span>
      <externalReferences>
        <externalRef resource="NomBank" reference="decline.01" />
      </externalReferences>
      <role id="pr9r1" semRole="AM-MNR">
        <!--serious-->
        <span>
          <target id="t44" head="yes" />
        </span>
      </role>
      <role id="pr9r2" semRole="A1">
        <!--economic-->
        <span>
          <target id="t45" head="yes" />
        </span>
      </role>
    </predicate>
  </srl>
  <timeExpressions/>
</NAF>
```

References

[Álvez *et al.*, 2008] J. Álvez, J. Atserias, J. Carrera, S. Climent, A. Oliver, and G. Rigau. Consistent annotation of eurowordnet with the top concept ontology. In *Proceedings of Fourth International WordNet Conference (GWC'08)*, 2008.

[Álvez *et al.*, 2012] J. Álvez, P. Lucio, and G. Rigau. Adimen-sumo: Reengineering an ontology for first-order reasoning. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 8(4):80–116, 2012.

[Atserias *et al.*, 2004] J. Atserias, L. Villarejo, G. Rigau, E. Agirre, J. Carroll, B. Magnini, and Piek Vossen. The meaning multilingual central repository. In *Proceedings of GWC*, Brno, Czech Republic, 2004.

[Baker *et al.*, 1997] C. Baker, C. Fillmore, and J. Lowe. The berkeley framenet project. In *COLING/ACL'98*, Montreal, Canada, 1997.

[Bizer *et al.*, 2009] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. Dbpedia-a crystallization point for the web of data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):154–165, 2009.

[Björkelund *et al.*, 2009] Anders Björkelund, Love Hafdell, and Pierre Nugues. Multilingual semantic role labeling. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task*, CoNLL '09, pages 43–48, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[Björkelund *et al.*, 2010] Anders Björkelund, Bernd Bohnet, Love Hafdell, and Pierre Nugues. A high-performance syntactic and semantic dependency parser. In *Proceedings of the 23rd International Conference on Computational Linguistics: Demonstrations*, COLING '10, pages 33–36, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[Bohnet, 2010] Bernd Bohnet. Very high accuracy and fast dependency parsing is not a contradiction. In *The 23rd International Conference on Computational Linguistics (COLING 2010)*, Beijing, China, 2010.

[Bollacker *et al.*, 2008] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. ACM, 2008.

[Bosma *et al.*, 2009] Wauter Bosma, Piek Vossen, Aitor Soroa, German Rigau, Maurizio Tesconi, Andrea Marchetti, Monica Monachini, and Carlo Aliprandi. Kaf: a generic semantic annotation format. In *Proceedings of the GL2009 Workshop on Semantic Annotation*, 2009.

[Burchardt *et al.*, 2005] Aljoscha Burchardt, Katrin Erk, and Anette Frank. A WordNet Detour to FrameNet. In *Proceedings of the GLDV 2005 GermaNet II Workshop*, pages 408–421, Bonn, Germany, 2005.

[Buscaldi and Magnini, 2010] Davide Buscaldi and Bernardo Magnini. Grounding toponyms in an italian local news corpus. In *Proceedings of 6th Workshop on Geographic Information Retrieval (GIR'10), ACM (2010)*, pages 15:1–15:5, Zurich, Switzerland, 2010.

[Cao *et al.*, 2008] Diego De Cao, Danilo Croce, Marco Pennacchiotti, and Roberto Basili. Combining word sense and usage for modeling frame semantics. In *Proceedings of The Symposium on Semantics in Systems for Text Processing (STEP 2008)*, Venice, Italy, 2008.

[Erk and Pado, 2004] Katrin Erk and Sebastian Pado. A powerful and versatile xml format for representing role-semantic annotation. In *Proceedings of LREC-2004*, Lisbon, 2004.

[Fellbaum, 1998] C. Fellbaum, editor. *WordNet. An Electronic Lexical Database*. The MIT Press, 1998.

[Fillmore, 1976] Charles J. Fillmore. Frame semantics and the nature of language. In *Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language and Speech*, volume 280, pages 20–32, New York, 1976.

[Giuglea and Moschitti, 2006] Ana-Maria Giuglea and Alessandro Moschitti. Semantic role labeling via framenet, verbnet and propbank. In *Proceedings of COLING-ACL 2006*, pages 929–936, Morristown, NJ, USA, 2006. ACL.

[Gonzalez-Agirre *et al.*, 2012a] Aitor Gonzalez-Agirre, Egoitz Laparra, and German Rigau. Multilingual central repository version 3.0. In *LREC*, pages 2525–2529, 2012.

[González-Agirre *et al.*, 2012b] Aitor González-Agirre, German Rigau, and Mauro Castillo. A graph-based method to improve wordnet domains. In *CICLING*, pages 17–28. Springer, 2012.

[Gurevych *et al.*, 2012] Iryna Gurevych, Judith Eckle-Kohler, Silvana Hartmann, Michael Matuschek, Christian M Meyer, and Christian Wirth. Uby: A large-scale unified lexical-semantic resource based on lmf. In *Proceedings of EACL*, pages 580–590, 2012.

[Izquierdo *et al.*, 2007] Rubén Izquierdo, Armando Suárez, and German Rigau. Exploring the automatic selection of basic level concepts. In *Proceedings of RANLP*, volume 7. Citeseer, 2007.

[Johansson and Nugues, 2007] Richard Johansson and Pierre Nugues. Using WordNet to extend FrameNet coverage. In *Proceedings of the Workshop on Building Frame-semantic Resources for Scandinavian and Baltic Languages, at NODALIDA*, Tartu, Estonia, May 24 2007.

[Kipper, 2005] Karen Kipper. *VerbNet: A broad-coverage, comprehensive verb lexicon.* PhD thesis, University of Pennsylvania, 2005.

[Klein and Manning, 2003] Dan Klein and Christopher D. Manning. Fast exact inference with a factored model for natural language parsing. In *In Advances in Neural Information Processing Systems 15 (NIPS*, pages 3–10. MIT Press, 2003.

[Laparra and Rigau, 2013] Egoitz Laparra and German Rigau. Impar: A deterministic algorithm for implicit semantic role labelling. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)*, pages 33–41, 2013.

[Laparra *et al.*, 2010] Egoitz Laparra, German Rigau, and Montse Cuadros. Exploring the integration of wordnet and framenet. In *Proceedings of the 5th Global WordNet Conference (GWC 2010), Mumbai, India*, 2010.

[Lee *et al.*, 2011] H. Lee, Y. Peirsman, A. Chang, N. Chambers, M. Surdeanu, and D. Jurafsky. Stanford's multi-pass sieve coreference resolution system at the CoNLL-2011 shared task. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, page 28–34, 2011.

[Lee *et al.*, 2013] Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*, pages 1–54, January 2013.

[Levin, 1993] Beth Levin. *English verb classes and alternations: A preliminary investigation*, volume 348. University of Chicago press Chicago, 1993.

[Magnini *et al.*, 2006] Bernardo Magnini, Emanuele Pianta, Christian Girardi, Matteo Negri, Lorenza Romano, Bartalesi Lenzi, and Rachele Sprugnoli. I-cab: the italian content annotation bank. In *LREC'06*, Genoa, Italy, 2006.

[Marcus *et al.*, 1993] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotation corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330, 1993.

[McCallum, 2002] Andrew K. McCallum. MALLET: A machine learning for language toolkit. `http://mallet.cs.umass.edu`, 2002.

[Navigli and Ponzetto, 2010] Roberto Navigli and Simone Paolo Ponzetto. Babelnet: Building a very large multilingual semantic network. In *Proceedings of the 48th annual meeting of ACL*, pages 216–225, 2010.

[Palmer *et al.*, 2005] Martha Palmer, Daniel Gildea, and Paul Kingsbury. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106, March 2005.

[Palmer, 2009] Martha Palmer. Semlink: Linking propbank, verbnet and framenet. In *Proceedings of the Generative Lexicon Conference*, pages 9–15, 2009.

[Pennacchiotti *et al.*, 2008] Marco Pennacchiotti, Diego De Cao, Roberto Basili, Danilo Croce, and Michael Roth. Automatic induction of FrameNet lexical units. In *Proceedings of EMNLP*, 2008.

[Raghunathan *et al.*, 2010] K. Raghunathan, H. Lee, S. Rangarajan, N. Chambers, M. Surdeanu, D. Jurafsky, and C. Manning. A multi-pass sieve for coreference resolution. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, page 492–501, 2010.

[Shen and Lapata, 2007] Dan Shen and Mirella Lapata. Using semantic roles to improve question answering. In *Proceedings of the Joint Conference on (EMNLP-CoNLL)*, pages 12–21, 2007.

[Shi and Mihalcea, 2005] Lei Shi and Rada Mihalcea. Putting pieces together: Combining framenet, verbnet and wordnet for robust semantic parsing. In *Proceedings of CICLing*, Mexico, 2005.

[Subirats and Petruck, 2003] Carlos Subirats and Miriam R.L. Petruck. Surprise: Spanish framenet! In *Proceedings of the International Congress of Linguists*, Praga, 2003.

[Suchanek *et al.*, 2007] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: A Core of Semantic Knowledge. In *WWW conference*, New York, NY, USA, 2007. ACM Press.

[Taulé *et al.*, 2008] Mariona Taulé, Maria Antònia Martí, and Marta Recasens. Ancora: Multilevel annotated corpora for catalan and spanish. In *LREC*, 2008.

[Tonelli and Pianta, 2009] Sara Tonelli and Emanuele Pianta. A novel approach to mapping framenet lexical units to wordnet synsets. In *Proceedings of IWCS-8*, Tilburg, The Netherlands, 2009.

[Toutanova *et al.*, 2003] Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of HLT-NAACL*, pages 252–259, 2003.