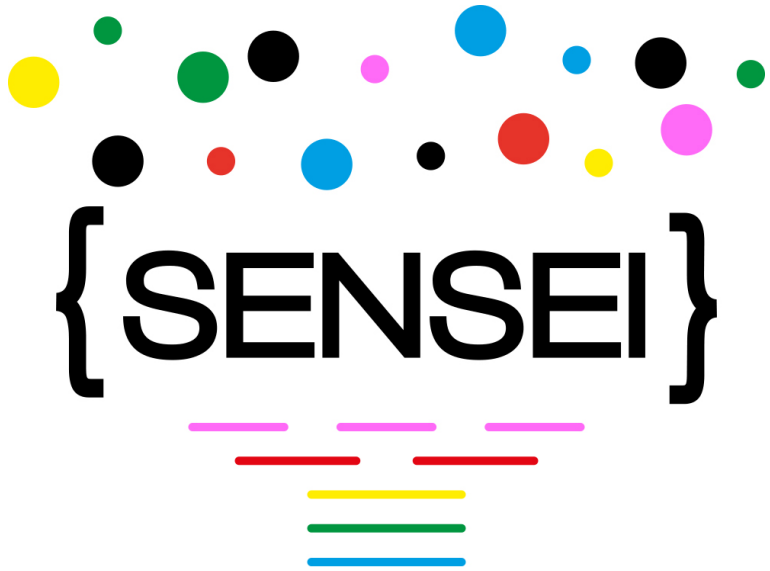


D3.2 – Report on the Multi-Domain and Cross-Media Parsing Model Adaptation

Document Number	D3.2
Document Title	Report on the Multi-Domain and Cross-Media Parsing Model Adaptation
Version	1.0
Status	Final
Work Package	WP3
Deliverable Type	Report
Contractual Date of Delivery	31.10.2015
Actual Date of Delivery	30.10.2015
Responsible Unit	UNITN
Keyword List	FrameNet, Domain Adaptation, Cross-Language Transfer
Dissemination level	PU

Editors

Evgeny A. Stepanov	(University of Trento, UNITN)
Frederic Bechet	(Aix Marseille Université, AMU)

Contributors

Evgeny A. Stepanov	(University of Trento, UNITN)
Ali Orkan Bayer	(University of Trento, UNITN)
Giuseppe Riccardi	(University of Trento, UNITN)
Benoit Favre	(Aix Marseille Université, AMU)
Frederic Bechet	(Aix Marseille Université, AMU)
Jeremie Tafforeau	(Aix Marseille Université, AMU)
Jeremie Trione	(Aix Marseille Université, AMU)
Mickael Rouvier	(Aix Marseille Université, AMU)

SENSEI Coordinator

Prof. Giuseppe Riccardi
Department of Information Engineering and Computer Science
University of Trento, Italy
giuseppe.riccardi@unitn.it

Document change history

Version	Date	Status	Author	(Unit)	Description
0.1	2015-07-31	Draft	E.A. Stepanov G. Riccardi	(UNITN)	Outline
0.1	2015-08-31	Draft	F. Bechet B. Favre J. Tafforeau J. Trione M. Rouvier	(AMU)	Outline updated Sections 2, 5 and 6 added
0.1	2015-08-31	Draft	A.O. Bayer	(UNITN)	Sections 3 and 4 added
0.1	2015-08-31	Draft	E.A. Stepanov G. Riccardi	(UNITN)	Outline updated
0.2	2015-09-01	Draft	A.O. Bayer	(UNITN)	Section 4 updated
0.2	2015-09-01	Draft	E.A. Stepanov	(UNITN)	Section 1 (introduction) updated
0.3	2015-09-26	Draft	E.A. Stepanov	(UNITN)	Section 1 (introduction) updated; Document structure updated
0.3	2015-10-03	Draft	A.O. Bayer	(UNITN)	Section 4 updated
0.4	2015-10-04	Draft	E.A. Stepanov	(UNITN)	Sections 1 (introduction), 3 and 4 updated; Section 7 (conclusion) added
0.5	2015-10-04	Draft	E.A. Stepanov	(UNITN)	Section 1 (introduction) updated; Executive Summary added
0.5	2015-10-12	Draft	E.S. Chiarani	(UNITN)	Quality check completed
0.6	2015-10-13	Draft	A.O. Bayer	(UNITN)	Section 4 updated
0.7	2015-10-15	Draft	E.A. Stepanov	(UNITN)	Formatting updates; Sections 3 and 4 updated;
0.8	2015-10-18	Draft	E.A. Stepanov A.O. Bayer	(UNITN)	Various minor updates and corrections;
0.9	2015-10-19	Draft	F. Bechet J. Trione J. Tafforeau	(AMU)	Update on semantic evaluation; Added semantic parsing to Section 6; biblio
1.0	2015-10-20	Draft	E.A. Stepanov	(UNITN)	Formatting updates; Overall revision;
1.0	2015-10-27	Final	E.S Chiarani G. Riccardi	(UNITN)	Final version;

Executive Summary

The objectives of WP3 is to automatically generate a structured semantic and para-semantic representation of human-human conversations in three SENSEI languages: English, French, and Italian. The three linguistic levels are addressed in the parsing process: (1) syntactic parsing to segment spoken dialogs or social media conversations into propositions; (2) Berkeley FrameNet semantic parsing to extract predicate/argument relations; and (3) para-semantic feature extraction of behavioral and emotional patterns, as well as sentiment polarities. The objective of Task 3.3, addressed by this deliverable, is to use of unsupervised or weakly supervised methods for adapting these parsing and feature extraction models from one application-domain or modality to another application-domain or modality. To this end we focus on using generic rich linguistic resources available in the three SENSEI languages in conjunction with cross-language and cross-domain adaptation methodologies.

The three adaptation approaches are addressed: cross-language adaptation via Statistical Machine Translation (Section 3), cross-domain adaptation through re-ranking of n-best lists of generic or in-domain parsers (Section 4), and cross-domain and cross-language adaptation of word embeddings (Sections 5 and 6). The first two approaches are addressed on the FrameNet semantic parsing task, the latter on the tasks of sentiment lexicon translation and also frame-semantic parsing.

We have observed that the cross-language adaptation with re-ranking methodology performs significantly worse than the in-domain semantic models. Moreover, the in-domain Italian semantic parser improves significantly with the re-ranking methodology. Therefore, we abandon the cross-language adaptation with re-ranking methodology and use the re-ranking methodology, in case any in-domain data is the desired language is available.

From the cross-language adaptation of word embeddings, we have observed that adapting an embedding space is as good as full-fledged SMT for sentiment lexicon translation. For the cross-domain word embedding adaptation, the proposed approach outperforms state-of-the-art Conditional Random Field approach on the frame-semantic parsing tagging task when little in-domain adaptation data is available.

Table of Contents

List of Acronyms and Abbreviations

Acronym	Meaning
AMT	Amazon Mechanical Turk
BPTT	Backpropagation Through Time
BWE	Bilingual Word Embeddings
CRF	Conditional Random Fields
HGI	Harvard General Inquirer lexicon
MPQA	Multi-Perspective Question Answering corpus/lexicon
NE	Named Entity
NLP	Natural Language Processing
NN	Neural Network
NRC	National Research Council (NRC) emotion lexicon
OOV	Out-of-Vocabulary
PoS/POS	Part-of-Speech
PWA	Per-Word Accuracy rate
RNN	Recurrent Neural Network
SLU	Spoken Language Understanding
SMT	Statistical Machine Translation
SVM	Support Vector Machines
WP	Work Package

1 Introduction

The objectives of WP3 is to automatically generate a structured semantic and para-semantic representation of human-human conversations in three SENSEI languages: English, French, and Italian. The three linguistic levels are addressed in the parsing process: (1) syntactic parsing to segment spoken dialogs or social media conversations into propositions; (2) Berkeley FrameNet semantic parsing to extract predicate/argument relations; and (3) para-semantic feature extraction of behavioral and emotional patterns, as well as sentiment polarities. The objective of Task 3.3 is to use of unsupervised or weakly supervised methods for adapting these parsing and feature extraction models from one application-domain or modality to another application-domain or modality. To this end we focus on using generic rich linguistic resources available in the three SENSEI languages in conjunction with cross-language and cross-domain adaptation methodologies.

Specifically, in this deliverable we report activities that took place during Period 2 of SENSEI project and whose objectives are making use of larger 'general' domain language *resources* (annotated or not) or *tools* trained on resource-rich languages to improve the performance of syntactic, semantic and para-semantic models for SENSEI data. In Section 2 we present the followed adaptation approaches, and in Sections 3, 4, 5 and 6 their applications to specific NLP tasks relevant to SENSEI.

The rest of this section presents how content of this deliverable connects to the Period 1 activities reported in the deliverable D3.1 (Section 1.1). Section 1.2 reports on how the reviewers' comments were addressed in Period 2. Section 1.3 specifically addresses the Recommendation 4

1.1 Follow-up to Period 1 Activities

During Period 1 the semantic models and the parsing methodology were developed in WP3 for processing the Human-Human conversations either using corpus-specific or generic state-of-the-art tools in other languages. The generic state-of-the-art tool in other language is SEMAFOR. In the Period 2, based on the reviewers' comments and the amount of effort needed to train in-language SEMAFOR models, the idea was abandoned and corpus specific parsers were trained.

SEMAFOR remained in the cross-language methodology. While in Period 1, there was no need for transferring its output to the source language, in Period 2 a word-alignment based annotation transfer was implemented to allow cross-domain re-ranking to take place. Thus, in Period 2, FrameNet parsing was extended with cross-language annotation transfer and cross-domain re-ranking.

1.2 Follow-Up to Recommendations from the First Review

Recommendation n.1:

"*Every language processing task, such as semantic role labeling, coreference resolution or summarization, should have a clear and formal definition, with a baseline given by the current state-of-the-art, and an upper bound of performance that can be expected.*"

We follow commonly accepted definitions of the task and subtasks. FrameNet parsing task is defined in Section 3. Specifically, for FrameNet semantic parsing we address identification of frames, i.e. detection of frame-triggering words and their classification into frames they trigger. Coreference resolution and summarization tasks are defined in WP4 and WP5, respectively.

For each task, the baselines are defined by the state-of-the-art or corpus-specific tools (e.g. LUNA FrameNet parser [12] and SEMAFOR [13]). When it comes to upper-bounds of performance, the common approach is to consider either inter-annotator agreement or oracle performances of n-best output. For domain adaptation with re-ranking we report the baseline and oracle performances (see Section 4).

Recommendation n.2:

"*A systematic error analysis, including the coverage analysis of the language processing algorithms, such as semantic role labeling, coreference resolution or summarization, and the categorization of errors, should be carried out in each task. Based on this analysis, the work should be prioritized.*"

Having analyzed the upper bound (oracle) of SEMAFOR with cross-language methodology, the approach is abandoned in favor of corpus-specific parsers.

Recommendation n.4:

"*Please consider re-implementing well-engineered state-of-the-art frame-semantic parsing in a clean architecture conforming to existing standards rather than adapting SEMAFOR, since the latter method involves significant engineering work.*"

Instead of re-implementing SEMAFOR to Italian and French, corpus-specific parsers were trained following the common approach across languages (Section 1.3). In Period 2, SEMAFOR remains only in cross-language adaptation methodology, and it is used *as-is*, without re-training any models. A comparison of the performance of the SENSEI architecture developed for English, French and Italian is described in the next section.

Recommendation n.5:

"*WP3 and WP4 should consider designing a joint processing architecture.*"

For Part-Of-Speech tagging, dependency parsing and Frame parsing a unified processing

pipeline has been defined. The same tools have been applied to English (FrameNet, Penn TreeBank), French (French TreeBank, Asfalda, RATP-DECODA) and Italian (Turin University Treebank, LUNA). This pipeline is used as a first step for WP4 coreference parser.

1.3 Evaluation of the SENSEI Pipeline for Semantic Frame Parsing

Following the reviewers' comments and the results obtained during the Period 1 of the project, we have developed a common pipeline for frame semantic parsing for the three SENSEI languages: English, French and Italian. This pipeline contains: (1) a syntactic parsing phase that performs Part-Of-Speech tagging, lemmatization and dependency parsing; then, (2) a semantic frame parser is applied on the obtained syntactic features. This frame parser is based on the *liblinear* library that implements linear separators. For each lexical unit (trigger), we have a classifier choosing a frame among all the possible ones. Once a frame has been selected, a frame-specific classifier is applied to finding the different frame elements and their roles.

For the frame classifiers, we use the following features:

- Lemma of the trigger
- POS of the trigger
- Syntactic path between the trigger and the frame elements
- Lemmas of the frame elements
- POS of the frame elements

For the frame element classifiers, we use:

- Syntactic path between the trigger and the frame elements
- Lemma of the frame element
- POS of the frame element
- Name of the trigger
- Trigger of the frame

In order to use the common pipeline for different languages and different application domains, we defined a *CoNLL* tabular format for encoding a text with all the syntactic features and semantic frame annotations.

Table 1: CoNLL tabular format for frame-semantic parsing.

id	Word	Lemma	PoS	GovId	syntLabel	Frame	roleInFrame1	roleInFrame2	...		
1	You	you	PRP	4	SBJ	_	Agent	_	_	Agent	Partner_1
2	and	and	CC	1	COORD	_	_	_	_	_	_
3	I	i	PRP	2	CONJ	_	_	_	_	_	Partner_2
4	have	have	VBP	0	ROOT	_	_	_	_	_	_
5	done	do	VBN	4	VC	Intentionnally_act	_	_	_	_	Undertaking
6	some	some	DT	8	NMOD	Relational_quantity	_	Denoted_quantity	_	_	_
7	important	important	JJ	8	NMOD	Importance	_	_	_	_	_
8	work	work	NN	5	OBJ	Working_on	Act	Mass	Factor	_	_
9	together	together	RB	5	MNR	Collaboration Manner	_	_	_	_	_

In the format, there are 7 columns that contain the syntactic features and the frame names (in case word is a trigger). The number of additional columns varies with respect to the number of frames in a sentence, i.e., if there are n frames (on 7^{th} column), there are n additional columns. The $7 + n^{th}$ column is the frame element role of the word in the n^{th} frame (see Table 1 for an example). The spans of multi-word frames were reduced to their syntactic heads.

This pipeline has been trained and evaluated on the 3 SENSEI languages, and compared to the publish state-of-the-art system results. The systems used for comparative evaluation are the following:

English	
SEMAFOR	trained and evaluated on the SemEval'07 FrameNet corpus (generic domain)
SENSEI (eng)	trained and evaluated on the SemEval'07 FrameNet corpus (generic domain)
French	
ASFALDA	trained on the French Tree Bank FrameNet corpus (generic domain) and evaluated on the DECODA corpus (specific domain)
SENSEI (fra)	trained and evaluated on the DECODA corpus (specific domain)
Italian	
LUNA	trained and evaluated on the Italian LUNA FrameNet corpus
SENSEI (ita)	trained and evaluated on the Italian LUNA FrameNet corpus

The comparative evaluation of the described systems is given in Table 2. As it can be observed, when dealing with specific domains (DECODA, LUNA), the SENSEI pipeline obtains good results, both for Frame and Role selection. For French, using a generic domain parser (ASFALDA) leads to inferior results than the application specific one. On the other hand, for English, the performance of the SEMAFOR parser is better than the SENSEI parser for generic domain. This can be explained by the fact that the SENSEI pipeline doesn't integrate any knowledge databases such as WordNet, and therefore is less prone to generalize on open domain data.

These results confirm the need for specialized parsers when dealing with specific domains

Table 2: Evaluation of the SENSEI FrameNet semantic parsing pipeline on 3 languages and 3 application domains. Comparison with stats-of-the-art baseline systems.

English	**Task**	**Precision**	**Recall**	**F-measure**
SENSEI (eng)	*Frame selection*	0.853	0.744	0.794
	Role selection	0.740	0.567	0.642
SEMAFOR	*Frame selection*	0.905	0.905	0.905
	Role selection	0.729	0.653	0.689
French	**Task**	**Precision**	**Recall**	**F-measure**
SENSEI (fra)	*Frame selection*	0.967	0.957	0.962
	Role selection	0.774	0.847	0.809
ASFALDA	*Frame selection*	0.913	0.897	0.905
	Role selection	0.656	0.418	0.510
Italian	**Task**	**Precision**	**Recall**	**F-measure**
SENSEI (ita)	*Frame selection*	0.677	0.561	0.614
	Role selection	0.405	0.249	0.308
LUNA	*Frame selection*	0.400	0.590	0.480
(Gold Frames)	*Role selection*	0.760	0.740	0.750

or specific languages (speech/chat). When enough annotated data is available, as for the DECODA corpus, training an application-specific semantic parser like the one developed in the SENSEI pipeline, provides good results. However it is not always possible to obtain enough annotated training data, therefore it is the goal of the Task 3.2 of Work Package 3 to develop strategies to adapt models to obtain such specialized parsers with less supervision. In the next Section 2 we describe the general adaptation approaches followed in SENSEI.

2 General Adaptation Approaches

Two kinds of 'conversations' are targeted in SENSEI: spoken conversations in customer service telephone call centers and text conversations represented by messages and comments in social-media platforms and web-chat applications. As much as these two kinds of conversations are different, they share the lack of in-domain genre specific NLP tools. Thus, both types of conversations require the adaptation of the existing tools and resources to their domains. In this section we briefly describe the adaptation approaches followed within the project.

2.1 Adaptation Methods

The followed adaptation approaches are roughly partitioned with respect to two aspects – used representation – neural or symbolic – and the 'field' of adaptation – language or domain.

Adaptation through Translation

English was the main focus of attention of the Natural Language Processing (NLP) community for years. As a result, there are significantly more annotated linguistic resources in English than in any other language. Consequently, data-driven tools for automatic text or speech processing are developed mainly for English. Developing similar corpora and tools for other languages is an important issue. However, this requires significant amount of effort. Statistical Machine Translation (SMT) techniques and parallel corpora were used to transfer annotations from a linguistic resource rich languages to a resource-poor languages for a variety of Natural Language Processing (NLP) tasks, including Part-of-Speech tagging, Noun Phrase chunking, dependency parsing, textual entailment, etc. The annotation transfer techniques are insensitive to the nature of 'annotations' – manual or automatic output of the NLP tools.

In this deliverable we recall the cross-language semantic parsing methodology defined in Period 1 of the project. Since the tools available in English are of general domain, the presented pipeline was completed with the annotation transfer techniques to the source language (from English to Italian) and domain adaptation (see Figure 1). The presented methodology is applicable to any NLP task cast as an identification of spans and their labeling.

Adaptation through Re-Ranking

Data-driven NLP techniques are very sensitive to the differences in training and testing conditions. Different domains and genres, such as news-wire written text and call-center conversation transcripts in Tech Support or Transportation domains, have different distributions of NLP task-specific properties; thus, the domain adaptation of the source language tools – either the

development of models with good cross-domain performance or tuning to the target domain – is critical.

In this deliverable we present the domain adaptation methodology based on re-ranking with in-domain semantic models – Conditional Random Fields (CRF) and Recurrent Neural Networks (RNN). The re-ranking is applied both to the output of the general domain semantic parsers through cross-language methodology and to the output of in-domain semantic parsers in the source language. The presented methodology is applicable in case in-domain data is available.

Adaptation in Low Dimension Vector-Space Models

Word embeddings have become ubiquitous in NLP, especially when using neural networks. One of the assumptions of such representations is that words with similar properties have similar representation, allowing for better generalization from subsequent models. In the standard setting, two kinds of training corpora are used: a very large unlabeled corpus for learning the word embedding representations; and an in-domain training corpus with gold labels for training classifiers on the target NLP task. Because of the amount of data required to learn embeddings, they are trained on large corpora of written text belonging to generic domains such as news or Wikipedia articles. This can be an issue when dealing with specific domain corpus or non-canonical language, such as spontaneous speech or social-media messages: embeddings have to be adapted to fit the particularities of specific domains/media.

However the adaptation corpus available for a given speech application can be limited, resulting in a high number of words from the embedding space not occurring in the adaptation space. We present in this deliverable a method for adapting an embedding space trained on a general purpose text corpus to a domain-specific corpus of limited size. In particular we deal with words from the embedding space not occurring in the adaptation data. We report experiments on a frame-semantic parsing task on spontaneous speech transcriptions collected in a call-center. We show that our word embedding adaptation approach outperforms state-of-the-art Conditional Random Field approach when little in-domain adaptation data is available.

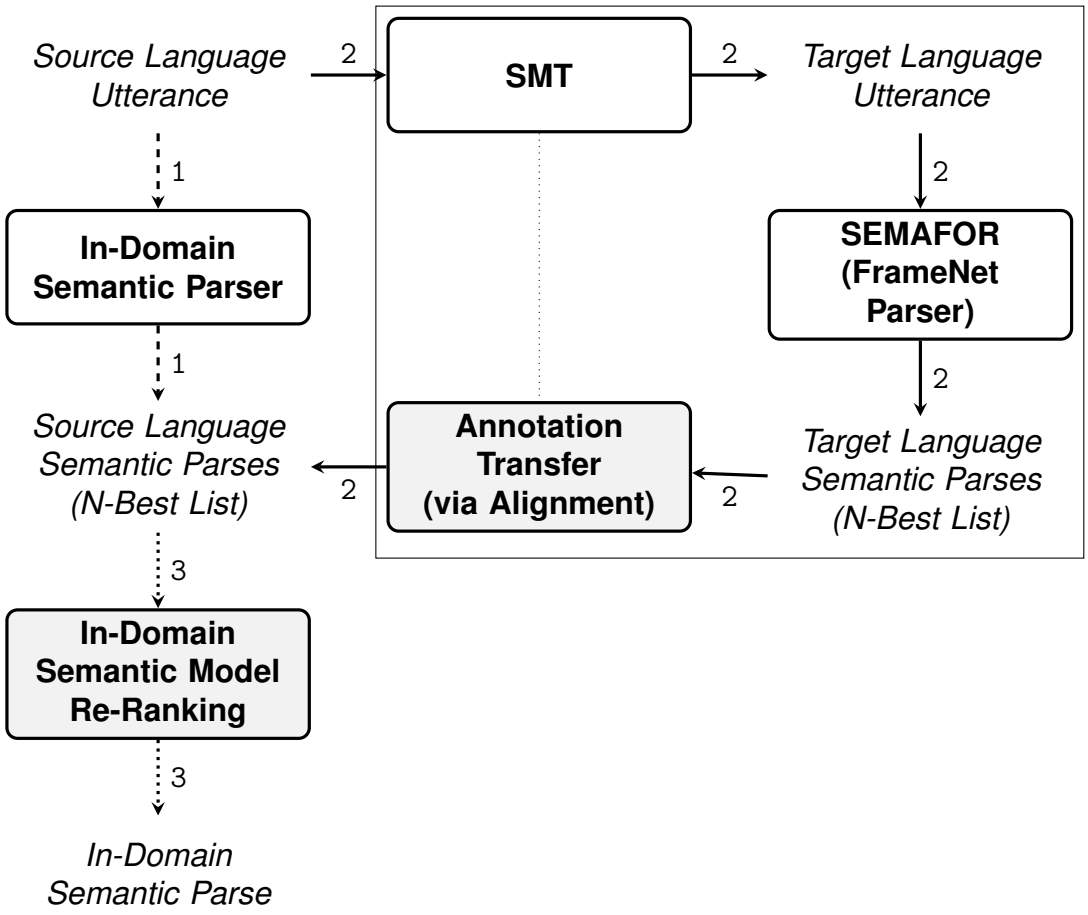


Figure 1: The cross-language and cross-domain adaptation pipeline (common to Sections 3 and 4). Cross-language adaptation (arrows with index 2 and boxed) consists of translation of the source language utterances to English, extraction of multiple semantic interpretations using SEMAFOR semantic parser in English. Cross-domain adaptation (dotted arrows with index 3) is a re-ranking of the semantic interpretations by in-domain semantic models in the source language (Italian). Semantic parsing with in-domain models is shown with dashed arrows with index 1. Components added in Period 2 of the project are in light-gray.

3 Cross-Language Adaptation via Statistical Machine Translation

In order to give the context to the cross-language adaptation of FrameNet semantic parsing, we first describe the semantic parsing task, repeating the description from the deliverable D3.1 (Period 1). Then we proceed to describe the cross-language methodology.

3.1 FrameNet Semantic Parsing

FrameNet semantic parsing is traditionally decomposed into the following sub-tasks:[1]

1. Identification of trigger words (target word detection), where the goal is to tag words that potentially trigger semantic frames. For instance, in “she declared to her friend that she was going out”. The target word *declared* is identified.
2. Classification of triggered words (target word labeling, frame disambiguation, frame identification), where the goal is to assign the relevant frame to the trigger word. For example, the sub-task assigns the frame STATEMENT to the trigger word *declared*.
3. Role filler identification (frame element detection, boundary detection), where the goal is to detect/segment the expressions that may fill a frame role (“she”, “to her friend” and “that she was going out” should be identified as potential role fillers.
4. Role filler classification (frame element labeling), where the goal is to assign the roles to the role fillers candidates. That is, “she”, “to her friend” and “that she was going out” are assigned the *Speaker*, *Addressee* and *Message* roles, defined for the frame STATEMENT, respectively.

The last two subtasks are generally referred to as “semantic role labeling” (SRL). However, the latter term is more general, as the task is not limited to the frame ‘roles’ defined in FrameNet. In SENSEI, however, we focus on the FrameNet frames and roles only.

In [12], the authors addressed the last two tasks on the Italian LUNA corpus, and demonstrated that the approach has satisfactory level of performance (see Table 3, the last two rows). However, automatic detection and labeling of trigger words has not. Thus, we address the first two tasks using cross-language methodology with SEMAFOR and compare it to the in-domain LUNA models.

[1] As terminology varies, in parentheses we provide alternative names for the sub-tasks.

Table 3: Performance on the LUNA FrameNet parser [12] on the parsing sub-tasks. Trigger detection and classification results are reported jointly. Role boundary detection performance is reported separately and jointly with role classification (i.e. its error is propagated to role classification step.

Task	P	R	F1
Trigger Detection			
+ Trigger Classification	0.40	0.59	0.48
Role Detection	0.89	0.86	0.87
+ Role Classification	0.76	0.74	0.75

3.2 Cross-Language Methodology

The cross-language adaptation with machine translation is applied to benefit from existing semantic parsers systems that are available in a language that is rich in resources for training semantic parsers, such as English. Through cross-language annotation transfer methodology, it is possible to transfer the semantic interpretation from the target language to the source language. However, since some semantic interpretation may be language specific and machine translation is noisy, a re-ranking mechanism should be utilized that uses semantic models in the source language.

The cross-language adaptation is performed by applying the following steps (see Figure 1: arrows with index 2):

1. Statistical Machine Translation (SMT) from the source language to the target language;
2. Semantic parsing in the target language by using the state-of-the-art semantic parser – SEMAFOR [13];
3. Transferring semantic annotation produced by the parser through statistical word alignment or a phrase table (produced at the SMT step);
4. Re-ranking of multiple hypotheses by an in-domain semantic model in the source language;

In Figure 1 the pipeline is divided into three segments:

1. default in-domain FrameNet semantic parsing done in the source language for comparison with the cross-language methodology (dashed arrows with index 1);
2. cross-language adaptation (arrows with index 2, boxed);
3. cross-domain adaptation through re-ranking (dotted arrows with index 3).

Table 4: Precision (**P**), recall (**R**), and F_1 (**F1**) of the trigger detection and classification performance of SEMAFOR through cross-language adaptation and in-domain LUNA semantic parser.

	P	**R**	**F1**
LUNA	0.40	0.59	**0.48**
SEMAFOR	0.27	0.28	0.27

In cases when the desired output is just a label and no re-ranking is intended, it is sufficient to apply Steps 1 and 2. This scenario was evaluated in Period 1 of the project and results are reported in the deliverable D3.1. Here we shortly remind the process and the obtained results.

For the SMT at Step 1, we have used an off-the-shelf translation system – Google Translate; and translated the source Italian utterances to English without training in-house SMT systems. For Step 3 – semantic parsing – the translated utterances are fed into the state-of-the art semantic parser (SEMAFOR). SEMAFOR is modified to output n-best list, which is re-ranked using a domain specific semantic model on the source language in Section 4.

In cases when the desired output is a span (labeled or not) or domain adaptation is intended, Step 3 is required. In Period 2 we trained statistical word alignments using GIZA++[43], and transferred the annotated output of SEMAFOR to Italian. Alternatively, it is possible to utilize phrase tables that are produced while training SMT systems using Moses [26].

3.3 Cross-Language Adaptation Results

In Period 1, we have carried out experiments on the frame identification with automatic triggers. The results are compared to the semantic parser of [12], which is a domain specific parser for Italian trained on LUNA corpus [14]. The second system is the state-of-the-art general purpose parser for English, SEMAFOR. SEMAFOR is evaluated by first translating Italian utterances to English and then running the parser on these translations. Table 4 presents the trigger detection and classification performance of these systems on LUNA Test Set after removing domain-specific specific frames (The Test Set for LUNA FrameNet annotation consists of 20 dialogs (1,146 turns) that contain 1,038 frames (145 unique). After removing the corpus-specific frames we are left with 958 frames (142 unique).)

From the table it is evident that in-domain data trained system outperforms SEMAFOR with cross-language methodology by 20 points. In the next section we describe the domain adaptation of these systems.

Table 5: Precision (**P**), recall (**R**), and F_1 (**F1**) of the *oracle* trigger detection and classification for SEMAFOR through cross-language adaptation and in-domain LUNA semantic parser.

	P	R	F1
LUNA	0.82	0.69	0.75
SEMAFOR	0.43	0.36	0.39

4 Cross-Domain Adaptation through Re-Ranking

The semantic parsing systems – LUNA [12] and SEMAFOR [13] were re-designed such that they output multiple semantic interpretation hypotheses, i.e. n-best list. In this section we present the domain-adaptation methodology that re-ranks these n-best lists to identify hypotheses that are closer to the domain. The re-ranking is performed using in-domain semantic models.

As a baseline of the method we consider 1-best system performances given in Table 4. The upper-bound in re-ranking, on the other hand, is given by the oracle of the generated n-best list (i.e. the best hypothesis among all available ones). The oracles of the systems are given in Table 5. Comparing the baseline results from Table 4 and the oracle performances, we already observe that in-domain baseline is higher than the oracle of SEMAFOR through cross-language methodology. Thus, in-domain and 'in-language' models are always preferable.

In the next sections, we first present the re-ranking methodology with semantic models that are trained on the source language. Then, we report the results of the re-ranking experiments of both SEMAFOR and the LUNA semantic parser.

4.1 Source-Language In-Domain Semantic Models

We have trained two different language-dependent semantic models on LUNA Human-Human corpus [14]. The first semantic model is based on Conditional Random Fields (CRFs) [29], which are successfully used in sequence labeling. The second semantic model is a neural network model that is based on Recurrent Neural Networks (RNNs) [19]. RNNs also have been applied to sequence labeling and they have also achieved significant improvements in language modeling [36].

The features used to train the semantic models are extracted by using TextPro Suit [15]. The features are: tokens, Part-of-Speech (PoS) tags, and Lemmas.

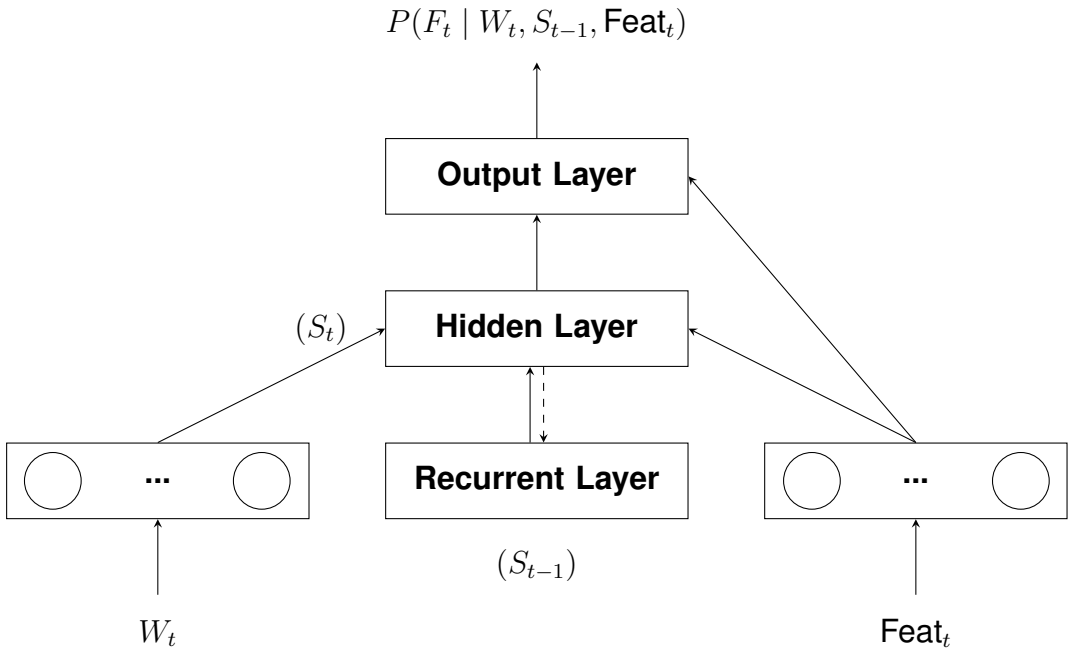


Figure 2: The RNN architecture of the semantic model. The network takes the current token and features (Part-of-Speech tags and lemmas) as 1-of-n encoding. It outputs the probability of each frame in the current context. The recurrent layer models a hidden state for the network.

CRF Semantic Model

The CRF model is built by using the CRF++ toolkit [28]. All the features are independent in the window of ± 1 tokens. The model is trained in a supervised way by using the reference frames on the training set of the LUNA Human-Human corpus.

RNN Semantic Model

The RNN semantic model is trained over the same features. The RNN semantic model has the structure that is depicted in Figure 2. It also uses maximum entropy features on word n-grams that are implemented as direct connections (not shown in the figure) with a hash-based implementation as given in [33]. PoS and lemma features also have direct connections to the output layer, however, n-gram features are not used over PoS and lemmas. The RNN semantic model outputs a probability distribution for frames given the features. It is trained by using backpropagation through time (BPTT) algorithm [8].

Table 6: The joint trigger detection and classification performances of the CRF and the RNN semantic models as precision (**P**), recall (**R**), and F_1 (**F1**). Performances of the LUNA parser and SEMAFOR are provided for comparison.

	P	**R**	**F1**
FrameNet Parsers			
LUNA	0.40	0.59	0.48
SEMAFOR	0.27	0.28	0.27
Semantic Models			
CRF	0.66	0.49	0.56
RNN	0.60	0.54	**0.57**

4.2 In-Domain Semantic Model Performances

The trigger detection and classification performance of the CRF semantic model and the RNN semantic model are given in Table 6. As can be seen from the results, the precision of these models is higher than the LUNA semantic parser that is presented in the previous section. However, these semantic models have lower recall. The main reason for that is these models do not use a separate target identification step, and miss more targets than the systems that implement a separate target detection step (e.g. [12]). We do not consider this a problem, since these models will be used to re-rank the multiple hypotheses that are generated by the systems that have a separate target detection step. Also, the F_1 of these models are better than the both LUNA Semantic parser and SEMAFOR via cross-language methodology.

4.3 Re-Ranking Experiments and Results

The re-ranking experiments are performed both on the output of LUNA semantic parser and of SEMAFOR. Although the re-ranking experiments are originally designed for cross-language adaptation, to compare both systems we perform re-ranking experiments with the LUNA semantic parser as well.

The re-ranking experiments on SEMAFOR are performed by first translating the utterances from Italian to English, this step is performed by using an off-the-shelf system, Google Translate. The translated utterances are fed into SEMAFOR and multiple hypotheses are obtained. The words in the translations are aligned to transfer the semantic interpretation. Finally, the hypotheses are re-ranked by using the semantic models that are presented in the previous section. The re-ranking experiments on the LUNA semantic parser, do not include any translation step. The re-ranking is performed on the multiple hypotheses that the LUNA parser outputs. The performance of the cross-language adaptation by re-ranking is given in Table 7.

Table 7: The joint trigger detection and classification performances of the domain-adaptation through re-ranking using the CRF and the RNN semantic models as precision (**P**), recall (**R**), and F_1 (**F1**).

	P	**R**	**F1**
LUNA Semantic Parser			
CRF	0.64	0.56	**0.60**
RNN	0.61	0.56	0.58
SEMAFOR			
CRF	0.33	0.23	0.27
RNN	0.32	0.23	0.27

As can be seen from the results, re-ranking the LUNA semantic parser improves the performance of the system significantly. Although there is a slight drop in the recall, the precision and the F_1 score increase significantly compared to the performance of the LUNA semantic parser. Performing re-ranking on SEMAFOR improves the precision from 0.27 to 0.32, however, the recall drops from 0.28 to 0.23. The F_1 score of SEMAFOR without re-ranking and with re-ranking are the same.

4.4 Conclusion and Period 3 Plans

The cross-language methodology with re-ranking does not perform well for the English semantic parser SEMAFOR. However, the Italian LUNA semantic parser improves significantly with the re-ranking methodology. Also, the re-ranked LUNA parser performs significantly better than the domain adapted SEMAFOR parser. Therefore, we abandon the cross-language adaptation with re-ranking methodology and use the re-ranking methodology for the LUNA semantic parser. In Period 3 of the project, the improved parser will be used for generating features for down-stream applications such as discourse parsing.

5 Cross-Language Adaptation with Vector-Space Models for Multi-Lingual Sentiment Analysis

In addition to semantic frame parsing, the Work Package 3 targets also the extraction of polarity and sentiment from text, especially short texts such as those left as comments or tweets for the SENSEI social-media use-case. Extracting such information automatically often implies having lexicons with polarity/sentiment labels. Similarly to frame, if this kind of resource is easily available for English, it is not the case for other languages with less linguistic resources.

Creating sentiment polarity lexicons is labor intensive. Automatically translating them from resource-rich languages requires in-domain machine translation systems, which rely on large quantities of bi-texts, not always available.

In order to overcome this resource issue, in D3.2, we propose to replace machine translation by transferring words from the lexicon through word embeddings aligned across languages through a simple linear transform. The approach leads to no degradation compared to machine translation, when tested on sentiment polarity classification on tweets from four languages.

5.1 Context of this Study

Sentiment analysis is a task that aims at recognizing in text the opinion of the writer. It is often modeled as a classification problem which relies on features extracted from the text in order to feed a classifier. Relevant features proposed in the literature span from microblogging artifacts including hashtags, emoticons [18, 3], intensifiers like all-caps words and character repetitions [27], sentiment-topic features [49], to the inclusion of polarity lexicons.

The objective of the work presented in this study is the creation of sentiment polarity lexicons. They are lists of word or phrase lists with positive or negative sentiment labels. Sentiment lexicons allow to increase the feature space with more relevant and generalizing characteristics of the input. Unfortunately, creating sentiment lexicons requires human expertise, is time consuming, and often results in limited coverage when dealing with new domains.

In the literature, it has been proposed to extend existing lexicons without supervision [2, 25], or to automatically translate existing lexicons from resourceful languages with statistical machine translation (SMT) systems [5]. While the former requires seed lexicons, the later are very interesting because they can automate the process of generating sentiment lexicons without any human expertise. But automatically translating sentiment lexicons leads to two problems: (1) out-of-vocabulary words, such as misspellings, morphological variants and slang, cannot

be translated, and (2) machine translation performance strongly depends on available training resources such as bi-texts.

In this study, we propose to apply the method proposed in [37] for automatically mapping word embeddings across languages and use them to translate sentiment lexicons only given a small bilingual dictionary. After creating monolingual word embeddings in the source and target language, we train a linear transform on the bilingual dictionary and apply that transform to words for which we don't have a translation.

We perform experiments on 3-class polarity classification in tweets, and report results on four different languages: French, Italian, Spanish and German. Existing English sentiment lexicons are translated to the target languages through the proposed approach, given gs trained on the respective Wikipedia of each language. Then, a SVM-based classifier is fed with lexicon features, comparing machine translation with embedding transfer.

5.2 Related Work

Many methods have been proposed for extending polarity lexicons: propagate polarity along thesaurus relations [16, 47, 20] or use co-occurrence statistics to identify similar words [55, 24].

Porting lexicons to other languages has also been studied: use aligned thesauri and propagate at the sense level [46, 17], translate the lexicon directly [22, 4], take advantage of off-the-shelf translation and include sample word context to get better translations [32] use crowd sourcing to quickly bootstrap lexicons in non-English languages [56].

5.3 Approach

Our approach consists in creating distributional word representations in the source and target languages, and map them to each other with a linear transform trained given a small bilingual dictionary of frequent words. Then, source language words from the polarity lexicon can be projected in the target language embedding. The closest words to the projecting are used as translation.

In our experiments, word embeddings are estimated on the source and the target language Wikipedia corpora using the word2vec toolkit [35]. The embeddings are trained using skip-gram approach with a window of size 7 and 5 iterations. The dimension of the embeddings is fixed to 200.

[30] have shown that the skip-gram word embedding model is in fact a linear decomposition of the co-occurrence matrix. This decomposition is unique up to a linear transformation. Therefore, given two word representations created from the same co-occurrence matrix, a linear transform can be devised to map words from the first to the second. Assum-

ing that co-occurrence matrices for the source and target languages are sampled from the same language-independent co-occurrence matrix, one can find a linear transform for mapping source words to target words, up to an error component which represents sampling error. This assumption is realistic for comparable corpora, such as embeddings trained on Wikipedia in various languages. However, word embeddings represent a mixture from the senses of each word, making the cross-language mapping non bijective (a word can have multiple translations), which will probably contribute to the residual. Therefore, it should be reasonable to train a linear transform to map words between the source and target languages. Note that a linear transform would conserve the translations associated to linguistic regularities observed in the vector spaces.

The idea is to translate words in another language in the goal to generate sentiment lexicon. In [37], the authors propose to estimate a transformation matrix W such that $Wx = y$, where x is the embedding of a word in the source language and y is the embedding of its translation in the target language.

In order to estimate the W matrix, suppose we are given a set of word pairs and their associated vector representations $\{x_i, y_i\}$ where x_i is the embeddings of word i in the source language and y_i is the embedding of its translation. The matrix W can be learned by the following optimization problem:

$$\min_W \sum_i || Wx_i - y_i ||^2 \quad (1)$$

which we solve with the least square method.

At prediction time, for any given new word x, we can map it to the other language space by computing $y = Wx$. Then we find the words whose representations are closest to y in the target language space using the cosine similarity as distance metric. In our experiments, we select all representations which cosine similarity is superior to λ (with $\lambda = 0.65$ set empirically).

In practice, we only have manual translations for a small subset of words, not necessarily polarity infused, on which we train W. We use that W to find translations for all words of the sentiment lexicon.

5.4 Experiments

5.4.1 Corpus and Metrics

The sentiment polarity classification task is set as a three-class problem: positive, negative and neutral. The metrics used to measure performance is macro-F-measure. We developed our system on French and apply the same components on Italian, Spanish and German. A concise description of the training data follows.

The French (FR) corpus comes from the DEFT'15 evaluation campaign [2]. It consists of 7,836 tweets for training and 3,381 tweets for testing. The Italian (IT) corpus was released as part of the SentiPOLC'14 evaluation campaign [6]. It consists of 4,513 tweets for training and 1,930 tweets for testing. For Spanish (ES), the TASS'15 corpus is used [48]. Since the evaluation campaign was still ongoing at the time of writing, we use 3-fold validation on the training corpus composed of 7,219 tweets. German (DE) tweets come from the Multilingual Sentiment Data Set [40]. It consists of 844 tweets for training and 844 tweets for testing.

In order to extract features on those corpora, polarity lexicons are translated from English using the method described in Section 5.3. The following lexicons are translated:

- **MPQA**: The MPQA (Multi-Perspective Question Answering) lexicon is composed of 4913 negatives words and 2718 positives words [57].
- **BingLiu**: This lexicon contains 2006 positive words and 4783 negative words. This lexicon includes mis-spellings, morphological variants and slang [23].
- **HGI**: The Harvard General Inquirer (HGI) lexicons contains several dictionaries, we only used positive and negative lexicons that contains respectively 1915 and 2291 words [53].
- **NRC**: NRC Emotion Lexicon is a large word list constructed by Amazon Mechanical Turk [39].

5.4.2 System

In order to test the value of the create lexicons, we use them in a typical sentiment polarity classification system [38]. We first tokenize the tweets with a tokenizer based on macaon [41]. Then, hashtags and usertags are mapped to generic tokens. Each tweet is represented with the following features and an SVM classifier with a linear kernel is trained to perform the task.

- Words n-grams
- All-caps: number of words with all characters in upper case
- Hashtags: number of hashtags
- Lexicons: number of words present in each lexicon
- Punctuation: number of contiguous sequences of exclamation marks, question marks, and both exclamation and question marks
- Last punctuation: whether the last token contains an exclamation or question mark

[2] https://deft.limsi.fr/2015/index.php

Table 8: Results in macro-F-measure obtained on the different languages (French, Italian, Spanish and German) using different sentiment lexicon (MPQA, BingLiu, HGI and NRC).

	FR	IT	ES	DE	All
No Sentiment Lexicon	65.83	58.20	59.79	52.84	60.65
[45]	65.97	-	-	-	-
[10]	65.22	57.98	60.61	53.55	60.95 ***(+0.30 pt)***
[31]	65.48	58.20	59.97	-	-
Moses (MPQA)	67.51	57.90	60.63	53.28	61.51 ***(+0.86 pt)***
Moses (BingLiu)	67.48	58.13	60.99	52.81	61.70 ***(+1.05 pt)***
Moses (HGI)	66.47	57.58	60.49	53.91	61.16 ***(+0.51 pt)***
Moses (NRC)	66.98	58.27	60.70	54.80	61.56 ***(+0.91 pt)***
BWE (MPQA)	67.38	58.35	60.61	53.24	61.53 ***(+0.88 pt)***
BWE (BingLiu)	66.87	58.25	60.63	52.26	61.33 ***(+0.68 pt)***
BWE (HGI)	66.33	58.14	60.61	55.00	61.34 ***(+0.69 pt)***
BWE (NRC)	66.62	58.31	60.39	56.88	61.45 ***(+0.80 pt)***
Moses + BWE (MPQA)	67.80	58.28	61.13	53.67	61.93 ***(+1.28 pt)***
Moses + BWE (BingLiu)	67.77	58.76	61.00	54.07	61.95 ***(+1.30 pt)***
Moses + BWE (HGI)	66.92	58.09	60.69	53.19	61.41 ***(+0.76 pt)***
Moses + BWE (NRC)	66.73	58.42	61.02	55.23	61.72 ***(+1.07 pt)***

- Emoticons: presence or absence of positive and negative emoticons at any position in the tweet
- Last emoticon: whether the last token is a positive or negative emoticon
- Elongated words: number of words with one character repeated more than three times, for example: “looooool”

We did not implement part-of-speech and cluster features as they cannot be assumed to be available in the target languages.

5.4.3 Results

Table 8 reports the results of the system and different baselines. The *No Sentiment Lexicon* system does not have any lexicon feature. It obtains a macro-F-measure of 60.65 on the four corpora. Systems denoted [45], [10], [31] are baselines that correspond respectively to unsupervised, supervised and semi-supervised approaches for generating the lexicon. We observe that adding sentiment lexicons improves performance. The *Moses* system consists in translating the different sentiment lexicons with the Moses SMT system. The approach based on translation obtains better results than the *Baseline* systems. The *BWE* (Bilingual

Word Embeddings) system consists in translating the sentiment lexicons with our method. This approach obtains results comparable to the SMT approach. Moses and BWE can be combined by creating a lexicon from the union of the lexicons obtained by those systems. This combination yields even better results than translation or mapping alone.

5.5 Conclusions

This study is focused on translating sentiment polarity lexicons from a resource-rich language through word embeddings mapped from the source to the target language. Experiments on four languages with mappings from English show that the approach performs as well as full-fledged SMT. While the approach was successful for languages close to English where word-to-word translations are possible, it may not be as effective for languages where this assumption does not hold. We will explore this aspect for future work.

6 Cross-Domain and Cross-Media Adaptation and OOV Handling with Vector-Space Models

Representation learning has emerged as a key issue in machine learning, and has led to major breakthroughs in computer vision and natural language processing [21, 52]. In particular researchers in NLP have focused on learning dense low dimensional (hundreds) representation space of words [35, 9], called embeddings, which model both semantic and syntactic information [34, 1]. The benefits of such representations is (1) that they offer a lower computational complexity when used as input of classifiers such as neural networks, and (2) that words with similar properties have similar representations, allowing for better generalization from subsequent models, e.g. for words not covered by targeted task training data.

In the standard setting of using an embedding space, two kinds of training corpora are used: a very large unlabeled corpus (C_{embed} for *Corpus Embeddings*) on which word representations are learned, and a smaller in-domain training corpus with gold labels for training classifiers on the target NLP task (C_{task}). It is assumed that C_{embed} has a much wider coverage than C_{task}, therefore all the words of C_{task} should have a representation in C_{embed}. This assumption is not always true when the in-domain data are very specific to a given context, or represent a different register of language than the standard canonical written language (e.g. Wikipedia) covered by C_{embed}.

Given a test corpus, some words might not have a representation extracted from C_{task}, forcing the classifier to rely on other features for making its decisions, often leading to mistakes. In this study, we aim at finding better representations for those Out-Of-Vocabulary (OOV) words, in order to limit their impact on subsequent tasks for both cross-domain and cross-media adaptation. We devise three types of OOVs: words covered by the task training corpus but not by the embedding corpus, words covered by the embedding corpus but not by the training corpus, and words covered by neither corpora.

This study presents a method that addresses these issues by both adapting an embedding space thanks to a small adaptation corpus, for a specific task, then by generalizing this adaptation to all words of the original embedding space, in particular to those not occurring in the adaptation corpus. Our contributions are as follows:

- Integration of word embeddings in a neural network performing a target task (here, semantic frame tagging)
- Embedding adaptation for words of the target task corpus through *refinement* (initialization of a hidden layer with original embeddings before training the neural network)

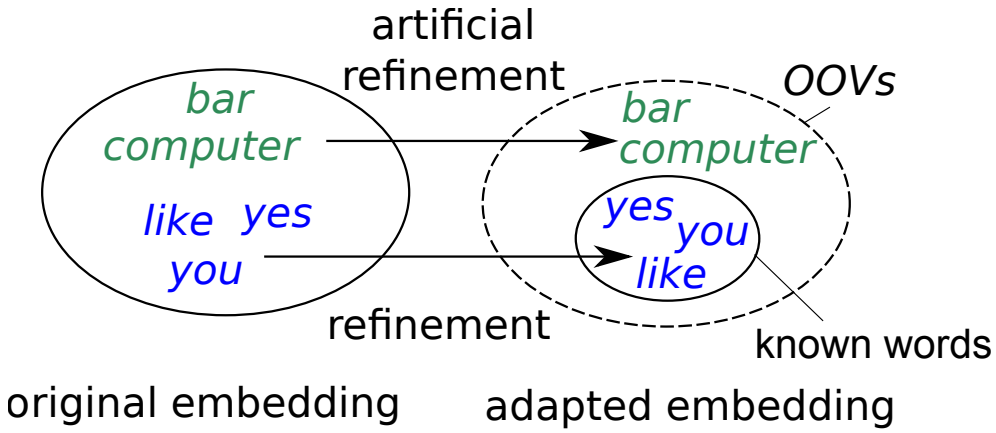


Figure 3: Illustration of the proposed adaptation process

- *Artificial refinement* for those out-of-vocabulary (OOV) words unseen in the target task training data.

The application framework of this study is the lightly supervised adaptation of a semantic frame tagger to process spontaneous speech transcriptions with very small amount of annotated training data and a very large unlabeled text corpus with a language-type mismatch (written text *v.s.* spontaneous speech). The proposed approach is illustrated in Figure 3. We show that our adaptation strategy improves over a state-of-the-art baseline using a CRF tagger when small amount of data is available to train the models, and when there is a mismatch between C_{embed} and the target corpus.

6.1 Related Work

OOV word handling in NLP tasks is dependent on the feature space used to encode data. Features can be computed from the sequence of characters composing the word (e.g. morphological, suffix and prefix features [51, 50]) in order to steer the classifier's decision when the form is unknown. Contextual features try to take advantage of the words in the vicinity of the OOV, such as n-grams in sequence models; contexts can be gathered in external corpora or using web queries [54]. OOVs can also be replaced by surrogates which have the same distributional properties, such as word clusters which have proved to be effective in many tasks [44]. Relaying on an embedding space for encoding words opens new possibilities for OOV handling: the availability of large corpora for learning embeddings and methods to process them [35] reduces the number of OOVs. For words unknown from the task training corpus (C_{task}) but occurring in the embedding corpus (C_{embed}), a similarity distance in the embedding space can be used to retrieve the closest known words and use its characteristics. For words not in C_{embed}, a generic OOV model is used. These methods are reviewed and evaluated in [1]

on a dependency parsing task showing that a small performance gain can be obtained when little training data is available. We propose in this study to push forward these experiments by extending the embedding space for all kinds of OOVs, not just for those not in C_{task}.

6.2 Different Kinds of Out-Of-Vocabulary Words

One may encounter different kinds of OOV words in an NLP task when using two different training corpus as we do, one for learning embeddings C_{embed} and one for learning our model C_{task} (here a POS tagger), and a test corpus for the task at hand C_{test}. Three kinds of OOV words can be defined, as presented in the following table: for instance, an OOV_2 word occurs in C_{task} (and of course in C_{test}) but not in C_{embed}.

OOV	C_{embed}	C_{task}	C_{test}
$\textbf{OOV}_1$	$\notin$	$\notin$	$\in$
$\textbf{OOV}_2$	$\notin$	$\in$	$\in$
$\textbf{OOV}_3$	$\in$	$\notin$	$\in$

We call hereafter OOV_1 the words that do not occur in any training set: OOV_2 are words that do occur in C_{task} but not in C_{embed}; and OOV_3 are words that do occur in C_{embed} but not in C_{task}. These categories are illustrated by Figure 4.

6.3 OOV Handling Strategies

We developed different strategies for processing these three kinds of OOVs. Dealing with an OOV_2 word only requires to initialize its embedding. We propose to use the embedding of the closest word (nearest neighbour) that belongs to C_{task}. We discuss below in 6.3.1 the similarity we used to select that neighbour.

Dealing with an OOV_3 word is more problematic since we would like to know accurately what would have been its refined embedding $\Phi_r(w)$ if it had occurred in C_{task}. We propose to approximate this refinement from the refinement of similar words in C_{task}. The underlying idea is that the refinement step shall smoothly transform the embedding space and that it may be well approximated locally. In other words, points that are close in the original embedding space will undergo a similar transformation. To artificially refine an OOV_3 word embedding we propose to apply to its original embedding $\Phi_0(w)$ the average transformation of its nearest neighbors. This processing is described in 6.3.2.

Finally we deal with OOV_1 words by successively applying the two processing steps above, finding an initial embedding and applying an artificial refinement to it.

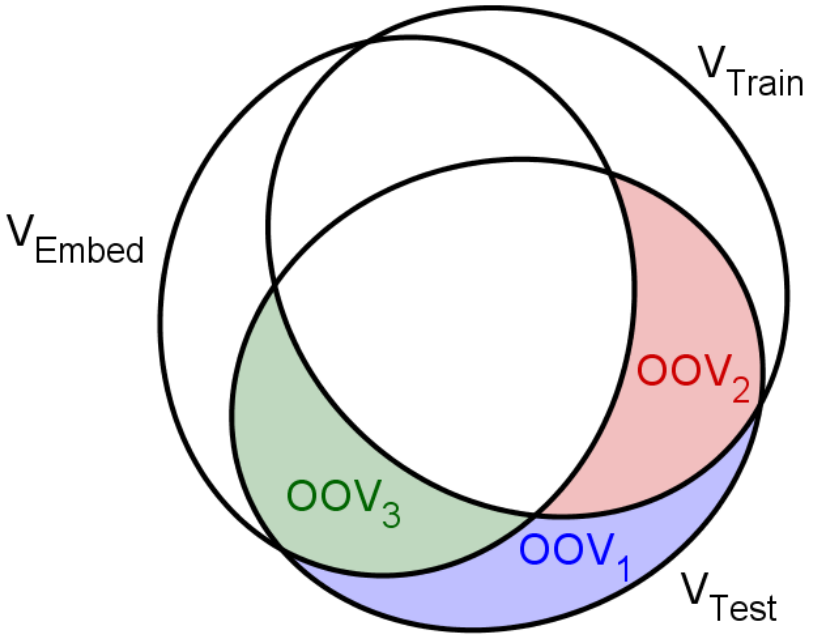


Figure 4: Different kinds of Out-Of-Vocabulary words

6.3.1 Setting an Initial Embedding

In this section, we consider words w which belong to C_{test} but which do not belong to the embedding learning corpus (OOV_{1+2}). $w \notin C_{\text{emb}}$ means we cannot provide any proper representation encoded into a regular embedding. This lead to errors even when considering words actually occurring in C_{task}. Setting an initial embedding for words $w \notin C_{\text{emb}}$ could be done by using a unique representation for each of them, either fixed *a priori*, or learned from low frequency words in the C_{emb} corpus[11]. An other option is to assign an individual embedding randomly initialized to each $w \notin C_{\text{emb}}$. We will use this strategy as a baseline in our experimental results.

We propose to estimate a relevant embedding representation for each word $w \notin C_{\text{emb}}$ with the following method:

- retrieving all occurrences of w in C_{task} or C_{test}
- finding the *closest* word t of w in C_{emb} thanks to all the context of occurrence of w
- replacing the unknown embedding of w by the one of t

The closeness between two words is defined according to the similarity between two distributions, one for each word, which represent the empirical distribution of occurrence of the word in all possible contexts (set of K previous words c_p and of K following words c_f).

More formally, we consider a word w and all of its occurrences in all possible contexts as the distribution of n-grams centered on w, where $n = 2K + 1$. This distribution is defined as $\left\{P_w(c_p, c_f), \forall (c_p, c_f) \in C_{\text{task}}^{2K}\right\}$ with:

$$\forall (c_p, c_f) \in C_{\text{task}}^{2K} \; P_w(c_p, c_f) = P(\langle c_p, w, c_f \rangle | w) = \frac{count\langle c_p, w, c_f \rangle}{count\langle w \rangle} \tag{2}$$

The similarity between two words u and v is computed as the KL-divergence between the two corresponding distributions. At the end, the embedding of a word $w \notin C_{\text{emb}}$ is set to the embedding of its closest word $t = \underset{u \in C_{\text{emb}} \cap C_{\text{task}}}{argmin} \; D_{KL}(P_w || P_u)$.

$$D_{KL}(P_u || P_v) = \sum_{c_p, c_f} P_u(c_p, c_f) \; \log \frac{P_u(c_p, c_f)}{P_v(c_p, c_f)} \tag{3}$$

6.3.2 Artificial Refinement of an Embedding

To simulate the adaptation of an embedding through learning, i.e. to infer the refined embedding of a word t, $\Phi_r(t)$, we chose to compute the average transformation of the embedding of its K nearest neighbours in the original embedding space, $(n_k)_{k=1..K}$, and to apply it to $\Phi_0(t)$ yielding: $\Phi_r(t) = \Phi_0(t) + \sum_{k=1}^{K} \alpha_k (\Phi_r(n_k) - \Phi_0(n_k))$ where the mixing coefficients α are positive real values that sum to one and which are proportional to the similarity between t and n_k (cosine similarity in our experiments).

6.4 A Neural Network Framework for Semantic Frame Tagging

We use in this study the *RATP-DECODA* [3] corpus described in D2.2. It consists of 1514 conversations over the phone recorded at the Paris public transport call center over a period of two days [7]. The calls last 3 minutes on average, representing a corpus of about 74 hours of signal. The call center dispenses information and customer services, and the two-day recording period covers a large range of situations such as asking for schedules, directions, fares, lost objects or administrative inquiries.

[3] The RATP-DECODA corpus is available for research at the Ortolang SLDR data repository: `http://sldr.org/sldr000847/fr`

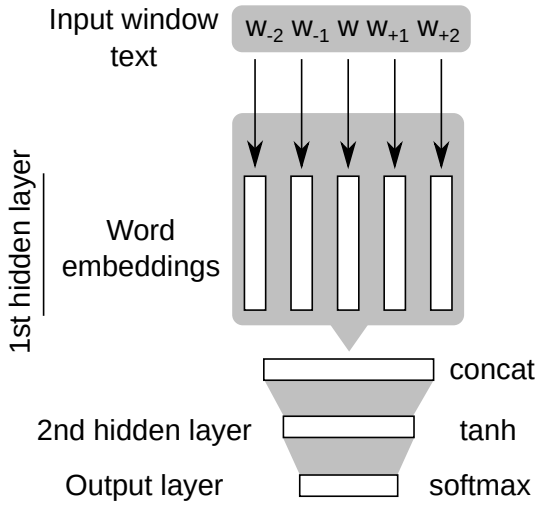


Figure 5: *Our system is a neural network which takes as input a window of words centered on the word to label. It is learned to predict the semantic frame label of the word of interest.*

The *RATP-DECODA* has been annotated with semantic frames as presented in deliverable D3.2. In our experiments, the semantic frame annotations are projected at the word level: each word is either labeled as `null` if it is not part of a frame realization, or as the name of the frame (or frame elements) it represents. In our corpus, 26% of the words have a non-null semantic label and there are 210 different frame labels. A lot of ambiguities come from the disfluencies which are occurring in this very spontaneous speech corpus.

We have shown in Section 1.3 that the SENSEI pipeline semantic frame parser obtains good results when enough training data is available. Our goal in this study is to evaluate our embedding adaptation strategy when little training data is available. In particular for all words missing in C_{emb} corpus. To do so we defined a simple Neural Network architecture that takes these adapted embeddings as input, and predict semantic frame labels for each word as output. In this network the input layer is a lookup layer (also called embedding), that we note Φ, which transforms a sequence of words $(w_1, ..., w_T)$ to a sequence of low dimensional vectors $(\Phi(w_1), ..., \Phi(w_T))$. The transformation Φ is initialized with the embedding learned in an unsupervised fashion using the approach in [35]. It is further fine-tuned during the supervised training of the neural net on the SLU task.

More concretely, the neural architecture we use is similar to [11] and is illustrated in Figure 5. It uses a two hidden layers network whose input is a window of 5 successive words in a sentence centered on the word to label. Its expected output is one of the 211 FrameNet tags.

The first layer is a *lookup* layer that replaces each word by its embedding representation. This layer is implemented as a concatenation of 5 parallel hidden layers of size 300, the dimension of the embedding space, these parameters stand for the word embeddings and can be fine-tuned during training on SLU targets. This first hidden layer is fully connected to a second

Table 9: *Distribution of words in the test corpus C_{test} according to the different training partitions. Of course, the sum of* ***OOV***$_{1+3}$ *and* ***OOV***$_2$ *words is a constant. As the number of words in the task training corpus $|C_{\text{task}}|$ increases, an increasing number of* ***OOV***$_{1+3}$ *words become* ***OOV***$_2$ *words.*

C_{task}	$\lvert C_{\text{task}}\rvert$	**OOV**$_{1+3}$	**OOV**$_2$
D_0	1,667	1250 — 5.24%	1261 — 5.28%
D_1	11,273	697 — 2.92%	1814 — 7.60%
D_2	23,752	498 — 2.09%	2013 — 8.43%
D_3	65,057	203 — 0.85%	2308 — 9.67%
D_4	151,910	203 — 0.85%	2308 — 9.67%
D_5	230,950	157 — 0.66%	2354 — 9.86%
D_6	311,400	140 — 0.59%	2371 — 9.93%
D_7	387,689	132 — 0.55%	2379 — 9.96%
D_8	477,729	120 — 0.50%	2391 — 10.01%
D_9	576,056	108 — 0.45%	2403 — 10.06%

nonlinear hidden layer (256 neurons in our experiments) which is itself fully connected to an output layer of 211 neurons (one neuron per semantic frame label). This model is learned with stochastic gradient descent using a log-likelihood criterion. We use dropout regularization with a firing rate $p = 0.5$.

6.5 Experiments

The two datasets used in our experiments are the French RATP-DECODA corpus (500K words) for the in-domain labeled training corpus and the French part of Wikipedia for the unlabeled C_{embed} corpus (357M words). The RATP-DECODA corpus [7] collected within the DECODA project is made of 1514 conversations over the phone recorded at the Paris public transport call center. We used the same train/test partition as described in [42]. The train section C_{task} contains 521K words and the test section C_{test} 25K words. In order to test our adaptation strategy with different sizes of adaptation corpus, we split C_{task} into 10 nested sections of similar size from D_0 to D_9. Globally the amount of OOV words decreases when the amount of training data increases, however each section has a different distribution among the three OOV categories as some OOV$_1$ words (not in C_{task} and C_{embed}) becomes OOV$_2$ (in C_{task} but not in C_{embed}). We will focus on the following on OOV$_{1+3}$ (*real* OOV as not present in the C_{task} corpus) and OOV$_2$ (words in C_{task} but with no embeddings on which our recovery strategy apply).

Our experimental results are presented in Table 10 and Figure 6. Four systems are compared:

- **CRF** is a state-of-the-art Conditional Random Field tagger using lexical context of 5 words for predicting the best sequence of FrameNet labels.

Table 10: *Comparative results when only small amount of training data is available (D_0, D_1) and when the full training corpus is used (D_9)*

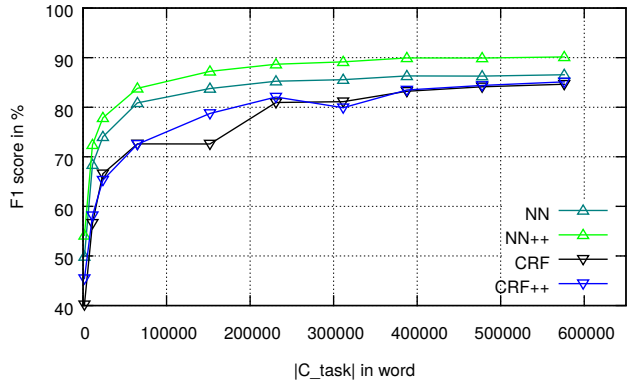
Train	Model	F1-score	PWA	**OOV**$_{1+3}$	**OOV**$_2$
D_0	CRF	40.16	78.64	83.20	86.76
	CRF++	45.44	80.67	92.96	87.87
	NN	49.90	80.15	76.88	62.01
	NN++	54.15	82.17	91.20	91.91
D_1	CRF	56.60	82.67	80.77	90.24
	CRF++	58.13	83.84	92.97	91.73
	NN	68.41	86.52	90.82	72.99
	NN++	72.43	88.22	92.11	93.61
D_9	CRF	84.63	92.36	81.48	94.09
	CRF++	85.12	92.85	97.22	95.30
	NN	86.56	93.13	94.44	81.15
	NN++	90.14	94.87	95.37	95.92

- **CRF++** is the same CRF using additional features (Part-Of-Speech, Named-Entities).
- **NN** corresponds to our Neural Network model described in Section 6.4 using a random initialization for unseen vectors as baseline.
- **NN++** integrates the word embeddings adaptation method proposed in Section 6.3.

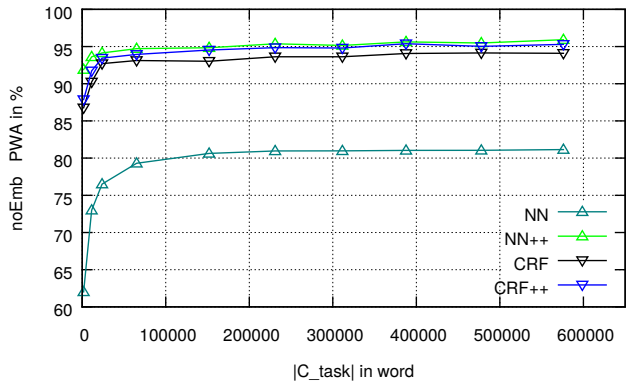
As we can see our strategy, which relies on a distributed representation of words to deal with OOV words, outperforms the CRF tagger, which had no access to external data. The gain is particularly significant when small amount of training data is available, but even when the full training corpus is used, we still observe improvements.

Adding POS and NE features improves performance (+1,25 F1-score on average for CRF++), especially for small corpora as it allows the CRF to generalize better on unseen data. Similarly we observe a very significant improvement from NN to NN+ by using our adaptation method. The embedding generation for words $w \notin C_{\text{emb}}$ leads to an average improvement of +3.34 F1-score. These results validate our adaptation approach.

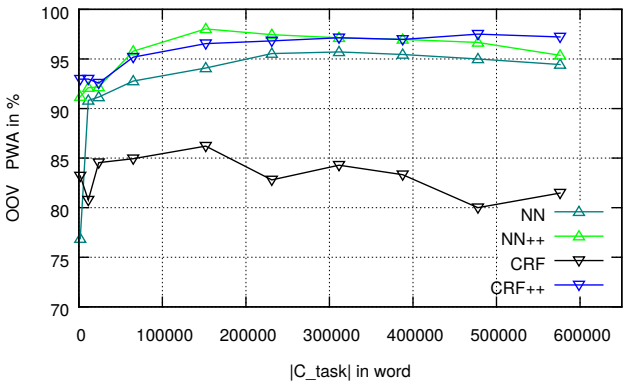
Focusing on the **OOV**$_2$ and **OOV**$_{1+3}$ accuracy, additional features increase the generalization of the subsequent models. POS and NE features help **OOV** recognition (6c) in the same way as embeddings adaptation fills the gap caused by mismatching resources (6b).



(a) F1 score



(b) **OOV**$_2$



(c) **OOV**$_{1+3}$

Figure 6: *F1-score and per-word accuracy rate (PWA, in %) restricted to* ***OOV***$_2$ *and* ***OOV***$_{1+3}$ *words as a function of the training corpus size. State-of-art baseline CRF tagger with and without additional features (POS, Named Entity) v.s. our proposed neural network model with and without adaptation strategy.*

6.6 Conclusion

The Section dealt with the particular problem of adapting a lexical embedding space to a specific SLU domain where a large number of application-specific terms do not have any representation in the initial vector space. We proposed to adapt lexical embeddings by creating accurate representations for unknown words: **OOV** words which do not occur in the SLU training data and words from the target domain which do not appear in the embedding training data. We showed on a semantic frame tagging task that our adaptation strategy improves over a state-of-the-art baseline using a CRF tagger when there is a mismatch between C_{emb} and the target corpus, especially when only a small amount of data is available to train the models.

7 Conclusions

At the end of Period 1, in the deliverable D3.1, we have presented the semantic models and the parsing methodology developed in WP3 for processing the Human-Human SENSEI conversations for social-media and speech data. The different methods for producing semantic representations were introduced from the three SENSEI languages – English, French, and Italian – either using corpus-specific or generic tools.

At the end of Period 2, in this deliverable, we have presented cross-languages adaptation methodology with subsequent cross-domain re-ranking for generic tools in the resource-rich languages like English and compared this methodology with the re-ranking of the output of corpus-specific semantic parser. As expected, the use of generic models with or without cross-language methodology produces lower performance. However, re-ranking of domain-specific models further improves the performance. Thus, using generic tools with cross-language methodology is left as the last resort in the case of absence of any in-domain annotated data. The work presented on adaptation in vector-space models, either cross-language or cross-domain, opens new pathways for other SENSEI task. In particular, uniform development of sentiment lexicons, that are used in speech summarization (WP5), will allow better comparison of the approaches.

In Period 3 of the project we plan to apply the adapted semantic parsers to discourse parsing in WP4 and summarization in WP5.

References

[1] Jacob Andreas and Dan Klein. How much do word embeddings encode about syntax. In *Proceedings of ACL*, 2014.

[2] Alina Andreevskaia and Sabine Bergler. Mining wordnet for a fuzzy sentiment: Sentiment tag extraction from wordnet glosses. In *EACL*, volume 6, pages 209–216, 2006.

[3] Sho Aoki and Osamu Uchida. A method for automatically generating the emotional vectors of emoticons using weblog articles. In *WSEAS*, pages 132–136, 2011.

[4] Carmen Banea, Rada Mihalcea, and Janyce Wiebe. Porting multilingual subjectivity resources across languages. *Affective Computing*, 4(2):211–225, 2013.

[5] Carmen Banea, Rada Mihalcea, Janyce Wiebe, and Samer Hassan. Multilingual subjectivity analysis using machine translation. In *EMNLP*, 2008.

[6] Valerio Basile, Andrea Bolioli, Malvina Nissim, Viviana Patti, and Paolo Rosso. Overview of the evalita 2014 sentiment polarity classification task. *EVALITA14*, 2014.

[7] Frederic Bechet, Benjamin Maza, Nicolas Bigouroux, Thierry Bazillon, Marc El-Beze, Renato De Mori, and Eric Arbillot. Decoda: a call-centre human-human spoken conversation corpus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*. European Language Resources Association (ELRA), may 2012.

[8] Mikael Bodén. A guide to recurrent neural networks and backpropagation. In *In the Dallas project, SICS Technical Report T2002:03, SICS*, 2002.

[9] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2787–2795. Curran Associates, Inc., 2013.

[10] Yanqing Chen and Steven Skiena. Building sentiment lexicons for all major languages. In *ACL*, pages 383–389, 2014.

[11] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *JMLR*, 12:2493–2537, 2011.

[12] Bonaventura Coppola, Alessandro Moschitti, and Giuseppe Riccardi. Shallow semantic parsing for spoken language understanding. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, NAACL-Short '09, pages 85–88, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[13] D. Das, D. Chen, A. F. T. Martins, N. Schneider, and N. Smith. Frame-semantic parsing. *Computational Linguistics*, 40(1):9–56, 2014.

[14] Marco Dinarelli, Silvia Quarteroni, Sara Tonelli, Alessandro Moschitti, and Giuseppe Riccardi. Annotating spoken dialogs: from speech segments to dialog acts and frame semantics. In *Proceedings of EACL Workshop on the Semantic Representation of Spoken Language*, Athens, Greece, 2009.

[15] Christian Girardi Emanuele Pianta and Roberto Zanoli. The textpro tool suite. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, may 2008. European Language Resources Association (ELRA).

[16] Andrea Esuli and Fabrizio Sebastiani. Pageranking wordnet synsets: An application to opinion mining. In *ACL*, volume 7, pages 442–431, 2007.

[17] Dehong Gao, Furu Wei, Wenjie Li, Xiaohua Liu, and Ming Zhou. Cross-lingual sentiment lexicon learning with bilingual word graph label propagation. *Computational Linguistics*, 2015.

[18] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, 1:12, 2009.

[19] Alex Graves. *Supervised Sequence Labelling with Recurrent Neural Networks*, volume 385 of *Studies in Computational Intelligence*. Springer, 2012.

[20] Ahmed Hassan, Amjad Abu-Jbara, Wanchen Lu, and Dragomir Radev. A random walk–based model for identifying semantic orientation. *Computational Linguistics*, 40(3):539–562, 2014.

[21] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580, 2012.

[22] Kanayama Hiroshi, Nasukawa Tetsuya, and Watanabe Hideo. Deeper sentiment analysis using machine translation technology. In *COLING*, page 494, 2004.

[23] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *SIGKDD*, pages 168–177, 2004.

[24] Nobuhiro Kaji and Masaru Kitsuregawa. Building lexicon for sentiment analysis from massive collection of html documents. In *EMNLP-CoNLL*, 2007.

[25] Hiroshi Kanayama and Tetsuya Nasukawa. Fully automatic lexicon expansion for domain-oriented sentiment analysis. In *EMNLP*, 2006.

[26] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics, 2007.

[27] Efthymios Kouloumpis, Theresa Wilson, and Johanna Moore. Twitter sentiment analysis: The good the bad and the omg! *Icwsm*, 11:538–541, 2011.

[28] Taku Kudo. CRF++. `http://taku910.github.io/crfpp/`, 2013.

[29] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289. Morgan Kaufmann Publishers Inc., 2001.

[30] Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In *NIPS*, pages 2177–2185, 2014.

[31] Isa Maks, Ruben Izquierdo, Francesca Frontini, Rodrigo Agerri, Piek Vossen, and andoni Azpeitia. Generating polarity lexicons with wordnet propagation in 5 languages. In *LREC*, 2014.

[32] Xinfan Meng, Furu Wei, Ge Xu, Longkai Zhang, Xiaohua Liu, Ming Zhou, and Houfeng Wang. Lost in translations? building sentiment lexicons using context based machine translation. In *COLING*, pages 829–838, 2012.

[33] T. Mikolov, A. Deoras, D. Povey, L. Burget, and J. Černocký. Strategies for training large scale neural network language models. In *Proceedings of ASRU*, pages 196–201. IEEE, 2011.

[34] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. In *Deep Learning Workshop at the 2013 Conference on Neural Information Processing Systems (NIPS)*, 2013.

[35] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[36] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernocký, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Interspeech 2010, 11th Annual Conference of the International Speech Communication Association*, pages 1045–1048, 2010.

[37] Tomas Mikolov, Quoc V Le, and Ilya Sutskever. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*, 2013.

[38] Saif M Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. In **SEM*, volume 2, pages 321–327, 2013.

[39] Saif M Mohammad and Peter D Turney. Nrc emotion lexicon. Technical report, NRC Technical Report, 2013.

[40] Sascha Narr, Michael Hulfenhaus, and Sahin Albayrak. Language-independent twitter sentiment analysis. *KDML, LWA*, pages 12–14, 2012.

[41] A. Nasr, F. Béchet, J.F. Rey, B. Favre, and J. Le Roux. Macaon: An nlp tool suite for processing word lattices. *Proceedings of the ACL 2011 System Demonstration*, pages 86–91, 2011.

[42] Alexis Nasr, Frederic Bechet, Benoit Favre, Thierry Bazillon, Jose Deulofeu, and Andre Valli. Automatically enriching spoken corpora with syntactic information for linguistic studies. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014), Reykjavik, Iceland, May 26-31, 2014.*, pages 854–858, 2014.

[43] Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, 2003.

[44] Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A Smith. Improved part-of-speech tagging for online conversational text with word clusters. In *HLT-NAACL*, pages 380–390, 2013.

[45] Alexander Pak and Patrick Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In *LREC*, 2010.

[46] Veronica Perez-Rosas, Carmen Banea, and Rada Mihalcea. Learning sentiment lexicons in spanish. In *LREC*, 2012.

[47] Delip Rao and Deepak Ravichandran. Semi-supervised polarity lexicon induction. In *EACL*, pages 675–682, 2009.

[48] Julio Villena Román, Eugenio Martínez Cámara, Janine García Morera, and Salud M Jiménez Zafra. Tass 2014-the challenge of aspect-based sentiment analysis. *Lenguaje Natural*, 54:61–68, 2015.

[49] Hassan Saif, Yulan He, and Harith Alani. Alleviating data sparsity for twitter sentiment analysis. In *MSM*, 2012.

[50] Cicero D. Santos and Bianca Zadrozny. Learning character-level representations for part-of-speech tagging. In Tony Jebara and Eric P. Xing, editors, *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1818–1826. JMLR Workshop and Conference Proceedings, 2014.

[51] Tobias Schnabel and Hinrich Schütze. FLORS: Fast and Simple Domain Adaptation for Part-of-Speech Tagging. *Transactions of the Association for Computational Linguistics (TACL)*, 2:15–26, February 2014.

[52] Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Y. Ng. Reasoning With Neural Tensor Networks For Knowledge Base Completion. In *Advances in Neural Information Processing Systems 26*, 2013.

[53] Philip Stone, Dexter C Dunphy, Marshall S Smith, and DM Ogilvie. The general inquirer: A computer approach to content analysis. *Journal of Regional Science*, 8(1):113–116, 1968.

[54] Shulamit Umansky-Pesin, Roi Reichart, and Ari Rappoport. A multi-domain web-based algorithm for pos tagging of unknown words. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pages 1274–1282. Association for Computational Linguistics, 2010.

[55] Leonid Velikovich, Sasha Blair-Goldensohn, Kerry Hannan, and Ryan McDonald. The viability of web-derived polarity lexicons. In *HLT*, pages 777–785, 2010.

[56] Svitlana Volkova, Theresa Wilson, and David Yarowsky. Exploring sentiment in social media: Bootstrapping subjectivity clues from multilingual twitter streams. In *ACL*, pages 505–510, 2013.

[57] Janyce Wiebe, Theresa Wilson, and Claire Cardie. Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, 39(2-3):165–210, 2005.