IJCLA ISSN 0976-0962

International Journal of Computational Linguistics and Applications

Vol. 6 No. 2 Jul-Dec 2015

Editor-in-Chief
ALEXANDER GELBUKH
Instituto Politécnico Nacional, Mexico

ISSN 0976-0962

International Journal of Computational Linguistics and Applications

Vol. 6 **No. 2** **Jul-Dec 2015**

International Journal of Computational Linguistics and Applications – IJCLA (started in 2010) is a peer-reviewed international journal published twice a year, in June and December. It publishes original research papers related to computational linguistics, natural language processing, human language technologies and their applications.

Indexing: Cabell's Directory of Publishing Opportunities.

Editor-in-Chief: Alexander Gelbukh

Subscription: Rs. 2999 / US$ 199

Payments can be made by Cheques/Bank Drafts/International Money Orders drawn in the name of BAHRI PUBLICATIONS, NEW DELHI and sent to:

BAHRI PUBLICATIONS
1749A/5, 1st Floor, Gobindpuri Extension,
Kalkaji, New Delhi 110019
Telephones: 011-65810766, (0) 9811204673, (0) 9212794543
E-mails: <bahrius@vsnl.com>; <bahripublications@yahoo.com>
Website: <http://www.bahripublications.in>

Printed & Published by **BAHRI PUBLICATIONS**, New Delhi

ISSN 0976-0962

International Journal of Computational Linguistics and Applications

Vol. 6 **No. 2** **Jul-Dec 2015**

CONTENTS

International Journal of Computational Linguistics and Applications, Vol. 6, No. 2, 2015, pp. 7–10
ISSN 0976-0962, http://ijcla.bahripublications.com

Editorial

This issue of IJCLA presents papers on cross-document event co-reference, web mining, statistical machine translation, lexical resources, question answering, metaphor detection, Twitter analysis, opinion mining,and assessing text complexity.

I. Pilán et al. (Sweden and Germany) propose a machine-learning approach to assess readability of Swedish texts for learners of this language as second language – that is, the language proficiency level required to understand the given text. They show that the most popular existing measure of readability used for Swedish as native language, LIX, is practically useless for assessing the language proficiency level required from learners of Swedish as second language, while their method distinguishes readability levels quite reliably.

A. Cybulska and **P. Vossen** (The Netherlands) introduce a robust approach to cross-textual event co-reference resolution and test it on a corpus of news articles. The approach compensates for the fact that even if the whole document, a news article in this case, is dedicated to a description of an event, the name of the event is not repeated several times in the text, which makes it difficult to detect automatically that the text is dedicated to this event. The authors use supervised classification of so-called sentence templates in order to detect co-reference of event mentions in different documents.

E. Chandrasekharan and **S. Chakraborti** (India) address the problem of gathering biographical information on people from the huge amount of traits left in Internet by digital interaction of the users. Gathering such information from the web can be both of a great historical, humanistic, and practical value and very dangerous when used maliciously. In either scenario, it is very important to understand what information and how can be collected from the web about each of us. The authors analyze the problem and give the results of their preliminary experiments.

M. Ammar and **S. Jamoussi** (Tunisia) consider decoding, the most important part of statistical machine translation algorithms. Decoding is an NP-complete problem and thus requires good heuristics for acceptable performance. The authors introduce a decoder based on artificial immune system-based metaheuristic algorithm. Evaluation, performed on two publicly available English-French corpora, shows that their approach is promising as compared with existing state-of-the-art decoders.

V. Nastase and **C. Strapparava** (Italy) argue for the importance of working with original lexical resources as opposed to resources artificially mapped to more standard resources such as WordNet, since in the latter case much of the information contained in the resource is lost, oversimplified, or altered by forceful and unnatural mapping. They present a machine-readable version of Wiktionary, a rich, and constantly growing, source of monolingual and cross-lingual lexical and semantic information about words. Unlike other authors who attempt to impose WordNet's sense inventory on Wiktionary, Nastase and Strapparava facilitate computational use of its own intrinsic structure.

C. Mărănducand **C. A. Perez** (Romania) present a Romanian dependency treebank, compiled by a combination of manual annotation and manually checked automatic annotation. The treebank will serve as a source of rules involving syntactic and semantic information on Romanian words, to be used in the construction of rule-based and hybrid dependency parsers for the Romanian language. The treebank is available in the universal dependency treebanks format for improved interoperability.

D. Clarke (UK) describes a simple and fast semantic parser based on a tensor product kernel. Semantic parsing is a sub-task of question-answering task; however, recently it tends to be understood as an approach to question answering that involves construction of a structured query from a natural-language question. The parser proposed by the author shows state-of-the-art performance while being much simpler in implementation and using less resources than existing state-of-the-art semantic parsers.

M. Mohler et al. (USA) present a methodology for detecting metaphors, both conventionalized and novel. The methodology uses supervised learning in a cross-lingual setting, so that metaphors in one language can be detected basing on the information learnt from another language. Metaphors are detected by generalization and analogical reasoning using semantic similarity, as well as transfer learning. An impressive performance of around 90% is achieved on English, Spanish, Russian, and Persian data.

V. Pinheiro et al. (Brazil) show how technology originally developed for tokenization of Chinese texts can be used for splitting composite hashtags used in social media, such as *#fergusondecision*, intro their component elements: *#*, *ferguson*, *decision*. This task is important in many natural language-processing techniques when they are applied to the succinct language of social media such as Twitter, where users give important information in the form of hashtags, which must be formally written as single words while in fact consisting of several words, such as a concept, idea, or a named entity.

L. Noce et al. (Italy) improve the performance of detection of anomalous user-generated context by combining text analysis with image analysis when the user-generated content includes images. Automatic detection of anomalous contents is a key element in fighting fake reviews in opinion mining: with such a technique, the companies such as TripAdvisor or Amazon, whole operation crucially depends on reliable user-generated reviews and on which we all, the users, crucially depend in making important decisions, can implement manual re-checking of suspicious user-generated contributions.

I. Pilán et al. (Sweden and Germany) propose a machine-learning approach to assess readability of Swedish texts for learners of this language as second language – that is, the language proficiency level required to understand the given text. They show that the most popular existing measure of readability used for Swedish as native language, LIX, is practically useless for assessing the language proficiency level required from learners of Swedish as second language, while their method distinguishes readability levels quite reliably.

This issue of IJCLA will be useful for researchers, students, software engineers, and general public interested in natural language processing and its applications.

ALEXANDER GELBUKH
EDITOR IN CHIEF

International Journal of Computational Linguistics and Applications, Vol. 6, No. 2, 2015, pp. 11–27
Received 25/01/2015, Accepted 25/02/2015, Final 18/11/2015.
ISSN 0976-0962, http://ijcla.bahripublications.com

"Bag of Events" Approach to Event Coreference Resolution. Supervised Classification of Event Templates

AGATA CYBULSKA
PIEK VOSSEN
Free University Amsterdam, Amsterdam

ABSTRACT

We propose a new robust two-step approach to cross-textual event coreference resolution on news articles. The approach makes explicit use of event and discourse structure thereby compensating for implications of the Gricean Maxim of quantity. News follows the principle of language economy. Information tends not to be repeated within discourse boarders. This phenomenon poses a challenge for models comparing information about event mentions (and their arguments) on the sentence level. Our approach addresses this challenge by building a knowledge representation per unit of discourse - for present purposes, a document. We collect event information from a single document filling in a "document template" and by that creating a "Bag of Events." We then use supervised Classification to determine if pairs of document templates contain corefering event mentions. Next we solve coreference between event mentions from the same document cluster by means of supervised classification of "sentence templates." The results indicate that the new approach is promising.

1. INTRODUCTION

Event coreference resolution is the task of determining whether two event mentions refer to the same event instance. This paper

explores cross-document resolution of coreference between events in the news.

It is common practice to use information coming from event arguments for event coreference resolution ([1], [2], [3], [4], [5], [6], [7], [8] among others). The research community seems to agree that event context information regarding time and place of an event as well as information about other participants play an important role in resolution of coreference between event mentions. Even though the contribution coming from event arguments as calculated in some studies does not directly translate into some significant increase of coreference resolution scores. [2] report that features related to event arguments slightly (+2.4% ECM F) improve intra-document event coreference. [7] note a ca. 4% CoNLL F-score improvement of within-topic event coreference resolution based on semantic similarity of event arguments.

Using entities for event coreference resolution is made complicated by the fact that descriptions of events at the sentence level often lack some pieces of information. As pointed out by [1], it could be the case however that a lacking piece of information might be available elsewhere within discourse boarders. News articles can be seen as a form of public discourse [9]. As such, the news follows the Gricean Maxim of quantity [10]. Journalists do not make their contribution more informative than necessary. This means that some information previously communicated within a unit of discourse, will not be mentioned again, unless pragmatically required. This is a challenge for models comparing mentions of events (and their arguments) with one another on the sentence level. One would like to be able to fully make use of information coming from event arguments. Instead of looking at event information available within the same sentence, we propose to take a broader look at event mentions surrounding the event mention in question within a unit of discourse. For the purpose of this study, we consider a document (here a news article) to be our unit of discourse.

This study experiments with an “event template” approach which employs the structure of event descriptions for event coreference resolution. In the proposed heuristic, event mentions

are examined through the perspective of five slots, as annotated in the dataset used in our experiments. The event slots correspond to different elements of event information: an event action (or an event trigger following the ACE terminology [11]) and four types of event arguments: time, location, human and non-human participant slots (see [12], whose ECB+ corpus [13] annotated with event coreference will be used in the experiments). The approach employed in this paper determines coreference between descriptions of events through compatibility of slots of an event template. Figure 1 presents an excerpt from topic 1, text number 7 of the ECB corpus [5]. Consider two event template examples presenting the distribution of event information over the five event slots in the two example sentences (Table 1).

The "American Pie" actress has entered Promises for undisclosed reasons. The actress, 33, reportedly headed to a Malibu treatment facility on Tuesday.

Figure 1. Text 7, topic 1, ECB corpus [5]

An event template can be filled from different units of discourse, such as a sentence, a paragraph or an entire document. We propose a two-step classification approach to event coreference resolution. In the first step of the approach, an event template is filled in per document; this is a "document template." By filling in a document template, one creates a "Bag of Events" per document. Bag of Events features are then used in supervised Classification.

This heuristic employs clues coming from discourse structure and namely those implied by discourse boarders. Descriptions of different event mentions occurring within a discourse unit, whether coreferent or related in some other way, unless stated otherwise, tend to share their context. In the example fragment (Figure 1) the first sentence reveals that an actress has entered a rehab facility. From the second sentence the reader finds out where the facility is located and when the actress headed there. It is clear to the reader of the example text fragment from Figure 1

that both event mentions from sentence one and two, happened on Tuesday. Also both sentences mention the same rehab center in Malibu. These observations are crucial for the Bag of Events approach proposed here. As the first step of the approach a document template is filled, accumulating mentions of the five event slots from a document, as exemplified in Table 1. Supervised classifiers determine whether pairs of document templates contain any corefering event mentions. In the second step of the approach coreference is solved between event mentions within document clusters created in step 1. For the purpose of this task again an event template is filled but this time, it is a "sentence template" which gathers event information from the sentence per action mention. Supervised classifiers solve coreference between pairs of event mentions and finally pairs sharing common mentions are chained into coreference clusters.

Table 1. *Sentence and document templates ECB topic 1, text 7, sentences 1 and 2*

Event Slot	**Sentence Template 1**	**Sentence Template 2**	**Document Template**
Action	*entered*	*headed*	*entered, headed*
Time	*N/A*	*on Tuesday*	*on Tuesday*
Location	*Promises*	*to a Malibu treatment facility*	*Promises, to a Malibu treatment facility*
Human Part.	*actress*	*actress*	*actress*
Non-human Part.	*N/A*	*N/A*	*N/A*

The main contribution of this work is the new robust Bag of Events approach to event coreference resolution that accounts for the realization of Gricean maxim of quantity in the news by incorporating Bag of Events features. The two-step Classification approach replaces the typically used in coreference resolution topic Classification step with document template Classification that allows for more specific event context disambiguation also within the same topic. Furthermore, the Bag of Events approach implies data representation through a relatively small number of features and yet delivers results comparable to those achieved in related work employing extended feature sets.

We will first delineate the Bag of Events approach to event coreference resolution in section 2. Section 3 reports on the experiments with the new method. We compare the results reached by means of our approach to those from related work in section 4. We conclude in section 5.

2. TWO-STEP BAG OF EVENTS APPROACH

We present a novel two-step approach to cross-textual event coreference resolution on news articles that explicitly employs event and discourse structure to account for implications of Gricean maxim of quantity. The first step in this approach is to build a knowledge representation by filling in an event template per unit of discourse - here a document. We collect all event action, location, time, human and non-human participant mentions from a single document and we fill in a document template (as depicted in Table 1). We then find pairs of document templates containing corefering event mentions by means of supervised Classification. In the second step, we use supervised classifiers to solve coreference between pairs of event mentions within clusters of document templates as determined in step 1. These steps are described in more detail below. Figure 2 depicts the implications of the two-step approach for the training data and Figure 3 for the test set.

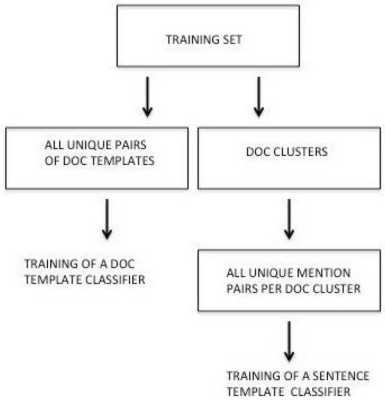


Figure 2. *Training set processing*

Figure 3. *Test set processing*

Step 1. Clustering document templates

The first step in this approach is to fill in a document template. We create a document template by collecting mentions of the five event slots: action, location, time, human and non-human participant from a single document. In a document template there is no distinction made between pieces of event information coming from different sentences of a document and no information is kept about elements being part of different mentions. A document template can be seen as a Bag of Events (and event arguments). The template stores a set of unique lemmas per event slot.

On the training set of the data, we train a pairwise binary classifier to determine whether two document templates share corefering event mentions. This is a supervised learning task in which we determine "compatibility" of two document templates if any two mentions from those templates were annotated in the corpus as coreferent. Let m be an event mention, and *doc* a collection of mentions from a single document template such that fm $\{m_i: 1 \leq i \leq doc_j\}$ where i is the index of a mention and J indexes document templates; doc_j: $1 \leq j \leq DOC$ where *DOC* are all document templates from the corpus. Let ma and mb be mentions from different document templates. "Compatibility" of a pair of document templates (doc$_j$; doc$_j$+1) is determined based on coreference of any mentions (ma_i, mb_i) from a pair of document templates such that:

$$coref\ (\exists ma_i \in doc_j,\ \exists mb_i \in doc_{j+1}) \Rightarrow compatibility\ (doc_j;\ doc_{j+1}).$$

On the training data, we train a binary decision-tree classifier (hereafter DT) to find pairs of document templates containing corefering event mentions.

After all unique pairs of document templates from the test set are classified by means of the DT document template classifier, "compatible" pairs are merged into document clusters based on pair overlap.

Step 2. Clustering sentence templates

The aim of the second step is to solve coreference between event mentions from document clusters which are the output of the Classification task from Step 1. We experiment with a supervised decision tree classifier. This time in the classification task pairs of sentence templates are considered.

A sentence template (e.g. Table 1) is created for every event action mention in the data set. All unique pairs of event mentions (and their sentence templates) are generated within clusters of documents sharing corefering event mentions in the training set. Pairs of sentence templates, that translate into features indicating compatibility across five event template slots, are used to train a decision tree sentence template classifier.

On the test set part of the data; after output clusters of the document template classifier from step 1 are turned to mention pairs (all unique action mention pairs within a document cluster), pairs of sentence templates are classified by means of the DT sentence template classifier. To identify the final equivalence classes of corefering event mentions, within each document cluster, event mentions are grouped into equivalence classes based on corefering pair overlap.

3. EXPERIMENTS

3.1. *Corpus*

For the experiments we used true mentions from the ECB+ corpus [13] which is an extended and re-annotated version of the ECB corpus [5]. The ECB+ corpus contains a new corpus component, consisting of 502 texts, describing different instances of event types that were already captured by the 43 topics of the

ECB. As recommended by the authors in the release notes, for experiments on event coreference resolution we used a subset of ECB+ annotations (based on a list of 1840 selected sentences), that were additionally reviewed with focus on coreference relations. Table 2 presents information about the data set used for the experiments. We divided the corpus into a training set (topics 1-35) and test set (topics 36-45).

3.2. *Experimental set up*

The ECB+ texts are available in the XML format. The texts are tokenized, so no sentence segmentation nor tokenization needed to be done. We POS-tagged (for the purpose of proper verb lemmatization) and lemmatized the corpus sentences. For the experiments we used tools from the Natural Language Toolkit ([14], NLTK version 2.0.4): the NLTK's default POS tagger, and WordNet lemmatizer[1] as well as WordNet synset assignment by the NLTK[2]. For machine learning experiments we used scikit-learn [15].

In the experiments, different features were assigned values per event slot (see Table 3). The lemma overlap feature (L) expresses a percentage of overlapping lemmas between two instances of an event slot, if instantiated in the sentence (with the exclusion of stop words). As the relation between an event and involved entities is not annotated in ECB+, frequently one ends up with multiple entity mentions from the same sentence for an action mention. All entity mentions from the sentence are considered in the overlap calculations. There are two features indicating event mentions' location within discourse (D), specifying if two mentions come from the same sentence and the same document. Action similarity (A) was calculated for a pair of active action mentions using the Leacock and Chodorow measure [16]. Per entity slot (location, time, human and non-human participant) we checked if there is any coreference relations annotated in the corpus between entity mentions from the

[1] www.nltk.org/ modules/nltk/stem/wordnet.html
[2] http://nltk.org/ modules/nltk/corpus/reader/wordnet.html

sentence for the two compared event actions; we used cosine similarity to express this feature (E). For all five slots a percentage of synset overlap is calculated (S). In case of document templates features referring to active action mentions were disregarded, instead only action mentions from a document were considered. All feature values were rounded to the first decimal point.

Table 2. *ECB+ statistics*

ECB+	#
Topics	43
Texts	982
Action mentions	6833
Location mentions	1173
Time mentions	1093
Human participant mentions	4615
Non-human participant mentions	1408
Coreference chains	1958

Table 3. *Features grouped into four categories: L-Lemma based, A-Action similarity, D-location within Discourse, E-Entity coreference and S-Synset based*

Event Slot	**Mentions**	**Feature Kind**	**Explanation**
Action	Active mentions	Lemma overlap (L) Synset overlap (S) Action similarity (A) Discourse location (D) - document - sentence	Numeric feature: overlap %. Numeric: overlap %. Numeric: [16]. Binary: - the same document or not. - the same sentence or not.
	Sent. or doc. mentions	Lemma overlap (L) Synset overlap (S)	Numeric: overlap %. Numeric: overlap %.
Location	Sent. or doc mentions	Lemma overlap (L) Entity coreference (E) Synset overlap (S)	Numeric: overlap %. Numeric: cosine similarity. Numeric: overlap %.
Time	Sent. or doc mentions	Lemma overlap (L) Entity coreference (E) Synset overlap (S)	Numeric: overlap % Numeric: cosine similarity. Numeric: overlap %.
Human Participant	Sent. or doc mentions	Lemma overlap (L) Entity coreference (E) Synset overlap (S)	Numeric: overlap %. Numeric: cosine similarity. Numeric: overlap %.
Non-Human Participant	Sent. or doc mentions	Lemma overlap (L) Entity coreference (E) Synset overlap (S)	Numeric: overlap %. Numeric: cosine similarity. Numeric: overlap %.

We experimented with a few feature sets, considering per event slot lemma features only (L), or combining them with other features described in Table 3. Before fed to a classifier, missing values were imputed (no normalization was needed for the scikit-learn DT algorithm). All classifiers were trained on an unbalanced number of pairs of document or sentence templates from the training set. We used grid search with ten fold cross-validation to optimize the hyper-parameters (maximum depth, criterion, minimum samples leafs and split) of the decision-tree algorithm.

3.3. *Baseline*

We will consider two baselines: a singleton baseline and a rule-based lemma match baseline. The singleton baseline considers event coreference evaluation scores generated taking into account all event mentions as singletons. In the singleton baseline response there are no "coreference chains" of more than one element. The rule-based lemma baseline generates event mention coreference clusters based on full overlap between lemma or lemmas of compared event triggers (action slot) from the test set.

Table 5 presents baselines' results in terms of recall (R), precision (P) and F-score (F) by employing the coreference resolution evaluation metrics: MUC [17], B3 [18], CEAF [19], BLANC [20], and CoNLL F1 [21]. When discussing event coreference scores must be noted that some of the commonly used metrics depend on the evaluation data set, with scores going up or down with the number of singleton items in the data [20]. Our singleton baseline gives us zero scores in MUC, which is understandable due to the fact that the MUC measure promotes longer chains. B3 on the other hand seems to give additional points to responses with more singletons, hence the remarkably high scores achieved by the baseline in B3. CEAF and BLANC as well as the CoNLL measures (the latter being an average of MUC, B3 and entity CEAF) give more realistic results. The lemma baseline reaches 62% CoNLL F1. A baseline only considering event triggers, will allow for an interesting comparison with our event template approach, employing event argument features.

3.4. *Evaluation*

Table 4 evaluates the final clusters of corefering event action mentions produced in the experiments by means of the DT algorithm when employing different features. The best coreference evaluation scores with the highest CoNLL F-score of 73% and BLANC F of 72% were reached by the combination of the document template classifier using feature set L across event slots and the sentence template classifier when employing features LDES (see Table 3 for feature de-scription). Adding action similarity (A) on top of LDES features does not make any difference on decision tree classifiers with a maximum depth of 5. Our best CoNLL F-score of 73% is an 11% improvement over the strong rule based event trigger lemma baseline, and a 34% increase over the singleton baseline.

Table 4. *Bag of Events approach to event coreference resolution, evaluated in MUC, B3, mention-based CEAF, BLANC and CoNLL F on the ECB+ corpus*

Step1			Step2			MUC			B3			CEAF	BLANC			CoNLL
Alg	Slot Nr	Feats	Alg	Slot Nr	Feats	R	P	F	R	P	F	F	R	P	F	F
-	-	-	DT	5	L	61	76	68	66	79	72	61	67	69	68	70
DT	5	L	DT	5	L	71	75	73	71	77	74	64	71	71	71	73
DT	5	L	DT	5	LDES	71	75	73	71	78	74	64	72	71	72	73
DT	2	L	DT	2	LDES	76	70	73	74	68	71	61	74	68	70	70
DT	5	L	DT	5	LADES	71	75	73	71	78	74	64	72	71	72	73

To quantify the contribution of document templates, we contrast the results of the two-step Bag of Events approach with scores achieved when skipping step 1 that is without the initial Classification of document templates. The results obtained with sentence template Classification only give us some insights into the impact of the document template Classification step. Note that the sentence template Classification without preliminary document template clustering is computationally much more expensive than the two-step template approach, which ultimately takes into account significantly less item pairs owing to the initial document template clustering. In the one-step approach the DT

sentence template classifier using lemma features (L), when trained on an unbalanced training set, reaches 70% CoNLL F. This is 8% better than the strong lemma baseline disregarding event arguments, but only 3% less than the two-step Bag of Events approach with the two classifiers trained on lemma features (L). The reason for the relatively small improvement by the document template classification step could arise from the fact that in the ECB+ corpus few sentences are annotated per text. 1840 sentences are annotated in 982 corpus texts, i.e. 1.87 sentence per text. We expect that the impact of document templates would be bigger if more event descriptions from a discourse unit were taken into account than only the ground truth mentions.

Table 5. *Baseline results: Singleton baseline and lemma match of event triggers evaluated in MUC, B3, mention-based CEAF, BLANC and CoNLL F*

Baseline	MUC			B3			CEAF	BLANC			CoNLL
	R	P	F	R	P	F	F	R	P	F	F
Singleton Baseline	0	0	0	45	100	62	45	50	50	50	39
Action Lemma Baseline	71	60	65	68	58	63	51	65	62	63	62

We ran an additional experiment with the four entity types bundled into one entity slot. Locations, times, human and non-human participants were combined into a cumulative entity slot resulting in a simplified two-slot template. When using two-slot templates for both, document and sentence classification on the ECB+ 70% CoNLL F score was reached. This is 3% less than with five-slot templates.

4. RELATED WORK

To the best of our knowledge, the only related study using clues coming from discourse structure for event coreference resolution was done by [1] who perform coreference merging between event template structures. Both approaches determine event compatibility within a discourse representation but we achieve that in a different way - with a much more restricted template

(five slots only) which facilitates merging of all event and entity mentions from a text as the starting point. [1] consider discourse events and entities for event coreference resolution while operating on the level of mentions.

Some of the metrics used to score event coreference resolution are dependent on the number of singleton events in the evaluation data set [20]. Hence for the sake of a meaningful comparison, it is important to consider similar data sets. The ECB and ECB+ are the only available resources annotated with both: within- and cross-document event coreference. To the best of our knowledge, no baseline has been set yet for event coreference resolution on the ECB+ corpus. Accordingly, in Table 6 we will also look at results achieved on the ECB corpus which is a subset of ECB+, and so the closest to the data set used in our experiments but capturing less ambiguity of the annotated event types [13]. We will focus on the CoNLL F measure that was used for comparison of competing coreference resolution systems in the CoNLL 2011 shared task.

Table 6. *The Bag of Events (BOE) approaches evaluated on ECB and ECB+ in MUC, B3, entity-based CEAF, BLANC and CoNLL-F in comparison with related studies. Note that the BOE approaches use gold and related studies system mentions*

Approach	Data	Model	MUC			B3			CEAF entity	BLANC			Co-NLL
			R	P	F	R	P	F	F	R	P	F	F
B&H	ECB annotation [5]	HDp	52	90	66	69	96	80	71	NA	NA	NA	NA
LEE	ECB annotation [6]	LR	63	63	63	63	74	68	34	68	79	72	55
BOE-2	ECB annotation [13]	DT+DT	65	59	62	77	75	76	72	66	70	67	70
BOE-5	ECB annotation [13]	DT+DT	64	52	57	76	68	72	68	65	66	65	66
BOE-2	ECB+ annotation [13]	DT+DT	76	70	73	74	68	71	67	74	68	70	70
BOE-5	ECB+ annotation [13]	DT+DT	71	75	73	71	78	74	71	72	71	72	**73**

The best results of 73% CoNLL F were achieved on the ECB+ by the Bag of Events approach using five slot event templates (*BOE-5* in Table 6). When using two-slot templates we get 3% less CoNLL F on ECB+. For the sake of comparison, we run an additional experiment on the ECB part of the corpus (annotation by [13]). The ECB was used in related work although with different versions of annotation so not entirely comparable. We

run two tests, one with the simplified templates considering two slots only: action and entity slot (as annotated in the ECB by [6]) and one with five-slot templates. The two slot Bag of Events (*BOE-2*) on the ECB part of the corpus reached comparable results to related works: 70% CoNLL F, while the five-slot template experiment (*BOE-5*) results in 66% CoNLL F. The approach of [6] (in Table 6 *LEE*) using linear regression (in Table 6 *LR*) reached 55% CoNLL F although on a much more difficult task entailing event extraction as well. The component similarity method of [7] resulted in 70% CoNLL F but on a simpler within topic task (not considered in Table 6). *B&H* in Table 6 refers to the approach of [5] using hierarchical Dirichlet process (*HDp*); for this study no CoNLL F was reported. In the *BOE* experiments reported in Table 6, during step 1 only lemma features L were used and for sentence template Classification (step 2) LDES features were employed. In all tests with the Bag of Events approach, ground truth mentions were used.

5. CONCLUSION

This paper presents a two-step Bag of Events approach to event coreference resolution. Instead of performing topic Classification before solving coreference between event mentions, as is done in most studies, this two-step approach first compares document templates created per discourse unit. Only after does it compare single event mentions and their arguments. In contrast to a heuristic using a topic classifier, that might have problems distinguishing between multiple instances of the same event type, the Bag of Events approach facilitates context disambiguation between event mentions from different discourse units. Grouping events depending on compatibility of event context (time, place and participants) on the discourse level, allows one to take advantage of event context information, which is mentioned only once per unit of discourse and consequently is not always available on the sentence level. From the perspective of performance, the robust Bag of Events approach using a small feature set also significantly restricts the number of compared items. Therefore, it has much lower memory requirements than a

pairwise approach operating on the mention level. Given that this approach does not consider any syntactic features and that the evaluation data set is only annotated with 1.8 sentences per text, the evaluation results are highly encouraging. Future research will be dedicated to experimenting with the Bag of Events approach on event slot mentions extracted by the system to demonstrate conclusively the validity of the approach.

REFERENCES

1. Humphreys, K., Gaizauskas, R. & Azzam, S. 1997. Event coreference for information extraction. In *ANARESOLUTION '97 proceedings of a Workshop on Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts*.
2. Chen, Z. & Ji, H. 2009. Event coreference resolution: Feature impact and evaluation. In proceedings of *Events in Emerging Text Types (eETTs) Workshop*.
3. Chen, Z. & Ji, H. 2009. Graph-based event coreference resolution. In *TextGraphs-4 Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing* (pp. 54-57).
4. Chen, B., Su, J., Pan, S. J., Tan, C. L. 2011. A unified event coreference resolution by integrating multiple resolvers. In proceedings of the *5th International Joint Conference on Natural Language Processing*, Chiang Mai, Thailand.
5. Bejan, C. A. & Harabagiu, S. 2010. Unsupervised event coreference resolution with rich linguistic features. In proceedings of the *48th Annual Meeting of the Association for Computational Linguistics*, Uppsala, Sweden.
6. Lee, H., Recasens, M., Chang, A., Surdeanu, M. & Jurafsky, D. 2012. Joint entity and event coreference resolution across documents. In proceedings of the *2012 Conference on Empirical Methods in Natural Language Processing and Natural Language Learning (EMNLP-CoNLL)*.
7. Cybulska, A. & Vossen, P. 2013. Semantic relations between events and their time, locations and participants for event coreference resolution. In proceedings of *Recent Advances In Natural Language Processing (RANLP-2013)*.
8. Liu, Z., Araki, J., Hovy, E. & Mitamura, T. 2014. Supervised within-document event coreference using information propagation. In proceedings of the *International Conference on Language Resources and Evaluation (LREC 2014)*.

9. van Dijk, T.A. 1988. News As Discourse. Routledge.
10. Grice, P. 1975. Logic and conversation. In P. Cole & J. Morgan (Eds.), *Syntax and Semantics* (pp. 41-58). New York: Academic Press.
11. LDC. 2005. ACE (Automatic Content Extraction) English Annotation Guidelines for Events ver. 5.4.3 2005.07.01. In *Linguistic Data Consortium*.
12. Cybulska, A. & Vossen, P. 2014. Guidelines for ECB+ annotation of events and their coreference. *Technical Report NWR-2014-1*, VU University Amsterdam.
13. Cybulska, A. & Vossen, P. 2014. Using a sledgehammer to crack a nut? Lexical diversity and event coreference resolution. In proceedings of the *International Conference on Language Resources and Evaluation (LREC 2014)*.
14. Bird, S., Klein, E. & Loper, E. 2009. *Natural Language Processing with Python*. O'Reilly Media Inc., http://nltk.org/book.
15. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. & Duchesnay, E. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.
16. Leacock, C. & Chodorow, M. 1998. Combining local context with wordnet similarity for word sense identification. In *WordNet: A Lexical Reference System and its Application*.
17. Vilain, M., Burger, J., Aberdeen, J., Connolly, D. & Hirschman, L. 1995. A model theoretic coreference scoring scheme. In *Proceedings of MUC-6*.
18. Bagga, A. & Baldwin, B. 1998. Algorithms for scoring coreference chains. In proceedings of the *International Conference on Language Resources and Evaluation (LREC)*.
19. Luo, X. 2005. On coreference resolution performance metrics. In proceedings of the *Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (EMNLP-2005)*.
20. Recasens, M. & Hovy, E. 2011. Blanc: Implementing the rand index for coreference evaluation. *Natural Language Engineering*, 17/4, 485-510.
21. Pradhan, S., Ramshaw, L., Marcus, M., Palmer, M.,Weischedel, R. & Xue, N. 2011. Conll2011 shared task: Modeling unrestricted coreference in ontonotes. In proceedings of *CoNLL 2011: Shared Task*.

ACKNOWLEDGMENTS

This work has been carried out within the NewsReader project supported by the EC within the 7th framework program under grant agreement nr. FP7-ICT-316404. We are grateful for the feedback from the anonymous reviewers. All mistakes are our own.

AGATA CYBULSKA
FREE UNIVERSITY AMSTERDAM
DE BOELELAAN 1105 1081HV AMSTERDAM
E-MAIL: <A.K.CYBULSKA@VU.NL>

PIEK VOSSEN
FREE UNIVERSITY AMSTERDAM
DE BOELELAAN 1105 1081HV AMSTERDAM
E-MAIL: <PIEK.VOSSEN@VU.NL>

International Journal of Computational Linguistics and Applications, Vol. 6, No. 2, 2015, pp. 29–42
Received 28/01/2015, Accepted 25/02/2015, Final 23/03/2015.
ISSN 0976-0962, http://ijcla.bahripublications.com

Footprints on Silicon: Explorations in Gathering Autobiographical Content

ESHWAR CHANDRASEKHARAN
SUTANU CHAKRABORTI
Indian Institute of Technology Madras, Chennai

ABSTRACT

As we interact with the world, we leave behind digital trails in the form of emails, blogs, tweets and posts, which serve as a rich source of data for generating our individual life stories, or autobiographies. Central to addressing the problem is the ability to discriminate content that is of autobiographical value from the rest. The features required for this classification task need to be discovered from the unstructured data, metadata, sentiments, properties of the social network and temporal properties of the interactions. In this paper we identify several dimensions of this problem, present some preliminary results on our explorations, and identify interesting research problems for the future.

1. INTRODUCTION

As more and more of human interactions occur online, a large amount of digital trails are left behind. One of the challenges that emerged in the UK Computing Research Committee's workshop on Grand Challenges for computing science in 2002, was entitled "*Memories for Life*." The idea was to analyze and use the digital data that people have about themselves which will soon be huge in size. An exemplar of this project was "*Stories from a Life*," which attempts to represent the stored memories in the form of stories. These stories could be generated under the supervision of the individual, or in an automated manner[1].

A systematic exploration of human interactions across the vastness of the Internet can serve as a rich source of data for generating individual life stories or autobiographies of people. Different sources like emails, blogs, tweets and posts on social networking sites can serve as potential sources of information about events of interest in a person's life. In this paper, we look at emails as the primary source of information and explore different methods to discriminate content that is of autobiographical value from the rest.

Emails are exchanged on a daily basis between several people, and contain varied types of content ranging from personal and professional messages to advertisements and spam. Over the course of a few years, there would be so much content buried in the inbox that correspond to important events or landmarks in the person's life. They could be about vacations, job promotions, cultivation of a new hobby, turning points in a person's life like the birth of new ones in the family and so on.

We are interested in building systems that analyze emails that get accumulated in a person's inbox over time, and identify those that could be of autobiographical value. We want to model the problem as a classification task, and try to use the unique structural properties of emails to gather autobiographical content from the collection of emails present in a person's inbox, in an automated manner.

Note that our system may not generate the summaries all by itself, but the gathered data will aid the user in creating one by prompting important details. The focus of our work is not the story generation part, but the crucial data gathering tasks that come before it.

2. DIMENSIONS OF THE PROBLEM

There are two methods that could be used by a person to generate an autobiography. The first is a top-down approach using cues like the person's hobby, biographical data, family members, and landmarks in personal and professional life to generate content. Research suggests that people use *interesting* events as "*anchors*" when trying to reconstruct memories of the past.

There has been work on probing the value of timelines and temporal landmarks for guiding search over subsets of personal content[2]. By using the landmark events that are identified by the person, a life-story can be created in top-down manner.

The second method is a bottom-up approach where we look at existing autobiographies and identify what discriminates events of autobiographical importance from the rest. We model the identification task as a classification problem, and are interested in looking at properties that could help in classifying autobiographical content present in a personal store of data, like a person's email inbox. We want to use a subset of emails that are labelled by the user and identify the features required for building our classifier. A summary of the gathered data can be used to generate the person's life story or autobiography with minimal supervision.

There are different dimensions which can be discovered from the unique structural properties of emails, in addition to textual content. We will look at some of these dimensions in detail, and look at how well they perform during classification.

2.1. *Text*

We would like to look at certain properties of the textual data present in emails exchanged by the user to aid our classifications.

Lexical features

We are interested in extracting textual keywords or subsets of words occurring in emails, that can describe the meaning of an email on the basis of properties such as frequency and length. We would like to look at the keywords present in an email and tell if it contains autobiographical content or not.

Language Function

Another interesting approach would be performing text classification using Language Function. The aim of Language Function Analysis (LFA) is to determine whether a text is predominantly expressive, appellative or informative. LFA is used to classify the predominant function of a text as intended by its author, and understand why a text was written[3]. It has been

observed that language functions relate to the writing style of a text, and they can be derived from statistical text characteristics obtained by using machine learning of lexical and shallow linguistic features like text type, writing style, sentiments and simple genre features.

Sentiment

The idea is that messages which contain high sentiments are likely to be expressing the user's opinion, stand or feeling about a particular topic in a conversation with another person through email, and are likely to be considered as containing information of autobiographical value by the user. Irrespective of the type of sentiment, negative or positive, an email displaying the user's sentiment regarding a topic is rich in content and could be classified as autobiographical.

2.2. *Mail network*

Our idea is that contacts with whom the user interacts on a one-to-one basis very frequently are generally close to the user, in a personal or professional nature. We could say that a contact is important if out of all the emails exchanged by the contact, most are one-to-one interactions with the user. This way we can compute the importance of a contact in a person's email network by looking at the sender-receiver characteristics of all the emails in the inbox.

2.3. *Email metadata*

Labels are assigned to each incoming email by the email client automatically, or manually by the user depending on different factors like the person sending the email, the words present in the subject and body of the email, and so on. Therefore, looking at the names of Labels or Filters that are assigned to an email can serve as a good indicator of the type of content present in an email, and could be used to distinguish between emails that contain autobiographical content and the rest.

2.4. *Time*

We would like to look at certain temporal properties of the user's interactions with other contacts to aid our classifications.

Threads

There are a lot of conversations or threads of emails present in a person's inbox where the user has exchanged emails with another contact or a group of contacts in succession. We would like to see if a large number of emails being exchanged between people on the same topic in a thread has an impact on the emails being of an autobiographical nature.

Burstiness

We could also look at the burstiness of emails being sent or received by a person, as an indication of the importance of emails. The periodicity, quantity and context shifts in the genre or type of content present in the emails being exchanged by the user can be used to identify mails containing autobiographical content. An example of a shift in context would be an email about travel tickets or say the birth of a baby, when all the previous mails were professional mails that were related to the person's work.

3. OUR APPROACH

We present a basic bottom-up scheme for performing the classification of emails containing autobiographical content by looking at different dimensions of emails, as our first work in a longer line of research. We build the classifier by mining discriminating features like textual keywords, threads, labels and mail network properties.

3.1. *Textual keywords*

The lexical features present in an email are obtained by tokenizing the textual data present in the "*Subject*" and "*Message Body*" components of the email. We perform keyword extraction on the textual content by removing stopwords, and using frequency of occurrence to rank the keywords. Then we perform

feature selection on the obtained keywords, by assigning scores to the different features using Information Gain values as the filter. Based on the scores, we rank the features and only keep the ones in the top which are the most distinguishing textual keywords in present in the email.

In Table 1, we list the top 50 words from the list of keywords that were obtained from the training set of user emails that gave the best performance when textual keywords were the only features used for building the classifier. These keywords can be used to quantitatively measure the lexical similarity of a new email with an email which is known to be tagged as containing autobiographical content or not.

Table 1. *Top 25 keywords from each class of emails obtained after feature selection using Information Gain filtering*

Autobiographical				
trip	undergraduate	going	tickets	booked
people	required	join	technology	feeling
confirmation	trekking	final	location	arrangements
places	finish	department	goal	inform
challenge	guidance	definitely	airport	dinner
Non-Autobiographical				
support	groups	stop	literature	subscribed
message	click	view	watch	reading
class	student	offers	matter	included
late	texts	encourage	bank	game
shop	anybody	read	story	software

3.2. *Mail network properties*

We compute the total number of emails exchanged by the user with the different contacts present in the user's contact list and also compute the number of these emails which are of a one-to-one nature with the user. A one-to-one email with the user is when there is only one recipient and one sender (which is always the case) and one of them is the user. Since we want to look at one-to-one interactions where the user is a primary part of the conversation, we make sure that the user's email ID is contained either in the "*to:*" or "*from:*" address in the email. We would not be interested in emails that are conversations between other

contacts and ignore emails where the user's email ID is contained in the "*cc:*" or "*Bcc:*" section of the email. We can use this information along with thread count of an email to capture the email network properties that are key to making a contact's email important to the user.

The features used to capture the email network properties are the number of overall mails sent by the e-mail's sender, number of recipients in the email and whether the user is a part of the email. Numerically,

Number of recipients = Number of email IDs present in "to:" address

We use the number of recipients as a numeric attribute to represent the number of contacts to which an email has been sent to. To see whether the user is a part of the email, we look at the email IDs of the sender and recipients to see if atleast one of them contains the user's ID.

We observed interesting properties in the mail network when we looked at the sender-receiver property along with the thread counts of emails. There were cases where we observed strong cycles being formed in the emails exchanged between the user and certain contacts. These were people to whom the user sent a one-to-one email and they responded back, and this cycle kept repeating for a number of times. We observed that these contacts were personal to the user and were of autobiographical value. There were other cases where emails were sent to groups of contacts, out of which just one or two would respond in a one-to-one manner with the user. We observed that these contacts were of a professional nature, where the interactions did not have as many cycles in their one-to-one interactions with the user, as observed in the previous case.

3.3. *Labels*

We collected the different label names that were present in our collection of emails and selected the most distinguishing labels. We observed that the labels that were most useful in the classification of emails containing autobiographical content were: "*Starred*", "*Important*", "*Sent*" and "*Inbox*".

These 4 label names were used as the metadata features for our classifications and we used them as f0,1g - features whose values are assigned based on whether the identified label names are assigned to an email or not.

3.4. *Threads*

We identify threads present in the inbox by looking at the "*Subject*" of emails. Emails that are part of the same thread contain the exact same "*Subject*" or with prefixes like "*Re:*" and "*Fwd:*" Numerically,

thread count = number of emails containing the same "Subject"

We compute thread counts for all the threads present in the Inbox and map the obtained values to the corresponding "*Subject*". We use this mapping to get the thread count feature for an email by using its "*Subject*" to perform a simple look-up. This gives us the size of the thread that the email is a part of, which is used as a numeric feature for our classification.

4. PRELIMINARY EXPERIMENTS

The corpus used for building and testing the model was obtained from the private emails of 3 different test users. Each user was asked to go through a representative set of emails present in his/her Inbox and tag them as containing autobiographical content or not. This kind of annotation helps us perform the task of gathering biographical content present in emails in a user-specific manner. It should also be noted that measures like inter-annotator agreement could not be computed as the emails used were private to a particular user.

From each user-tagged inbox, a random sample of 150 emails was used for the training phase and the classifications were done on another random sample of 80 test emails obtained from the user-tagged email inbox. We built 3 different classifiers, and report the average precision and recall values obtained for the 3 sets of user-tagged emails, when using di_erent combinations of features to build the classifier. We observed that

the average number of words present in the Email Corpus used for building our model to be 46746 words. Out of these, 3927 were distinct words, that were present in an English dictionary and were not stop words.

4.1. *Methodology*

The tagged email inbox of the test user is obtained in the form of a single *Mail Box* format file from the email client (Gmail, in our case). *Mail Box* (*MBox*) is a generic term for a family of related file formats used for holding collections of electronic mail messages. All of the messages present in a mail box are stored in a single. *MBox* text _le in a concatenated manner. We parse this file and separate the individual emails into different files of a similar format, to extract the features present in each email.

The typical information which we are interested in, that is contained in a *MBox* format file are the following:

- From
- Sent to
- Subject
- Metadata (labels, folders, date, time and other information)
- Body of the message

We conducted our experiments by using the following sets of features to obtain the vector representations of each email:

- Textual Keywords
- Email network properties
- Thread count
- Label or Folder name

4.2. *Results*

In order to reduce the bias in the choice of test and train data for building the classifier, we split the email corpora into 10 sets and run 10-fold tests using different train-test splits in the ratio 9:1. We compute average values of the obtained results for 3 different classifiers.

We used Naive Bayes, Random Forest and LibSVM classifiers in WEKA with default parameter values, to conduct our experiments on the tagged inbox of 3 test users using WEKA[4]. We look at the results that were obtained and compare the performance of different types of features and classifiers.

In Table 2, we have the results obtained for the different classifiers when using each of the individual type of features on their own. Here, P stands for Precision and R stands for Recall values that were obtained.

In Table 3, we have the results obtained when different combinations of features were used to build the classifiers. Here, *L* stands for Labels, *Txt* stands for Textual keywords, *Th* stands for Threads, *MN* stands for Mail Network, *All* stands for all 4 types of features, *P* stands for Precision and *R* stands for Recall values that were obtained.

Table 2. *Standalone performance of different types of features*

Classifier	Text		Labels		Threads		Mail Network	
	P	R	P	R	P	R	P	R
Naive Bayes	0.88	0.858	0.783	0.763	0.651	0.654	0.487	0.433
Random Forest	0.914	0.904	0.845	0.821	0.752	0.725	0.608	0.609
LibSVM	0.932	0.925	0.834	0.813	0.615	0.613	0.596	0.592

Table 3. *Performance of different combinations of features*

Classifier	L+Txt		L+Txt+MN		L+Txt+Th		All	
	P	R	P	R	P	R	P	R
Naive Bayes	0.868	0.846	0.874	0.854	0.879	0.854	0.888	0.867
Random Forest	0.948	0.946	0.942	0.942	0.952	0.95	0.951	0.95
LibSVM	0.952	0.95	0.959	0.958	0.908	0.904	0.921	0.917

We observed that the textual keywords and labels were most effective during classification when considered by themselves. Other features like email network properties and thread counts were not very good indicators on their own, but when augmented with textual keywords and labels, they were observed to give improved performances as seen in Figure 1.

We compute the average number of correctly classified instances observed for the 3 test user emails when using different

combinations of features and compare them to see how they differ, for the Naive Bayes classifier. In particular, we observe that text keywords and labels are strong indicators on their own. As a result, we consider the other 2 features: email network properties and thread counts on their own and see how the addition of labels and text keywords to these 2 sets of features impact the performance of the classifier, in a graphical manner in Figure 1. Here, *All* stands for all 4 features, *L* stands for Labels, *Txt* stands for Textual Keywords, *Th* stands for Threads, *MN* stands for Mail Network, *L+Txt+Mn* stands for combination of Labels, Textual Keywords and Mail Network, and so on.

A slight drop in performance was observed with the addition of thread count to the set of features in LibSVM, which was due to an increase in the number of misclassifications. This was due to the presence of certain test emails that had relatively high thread counts despite not having autobiographical content in them. Overall, the best performance was observed when all 4 types of features were used to build the classifier and the most discriminating individual features were observed to be text keywords and label names.

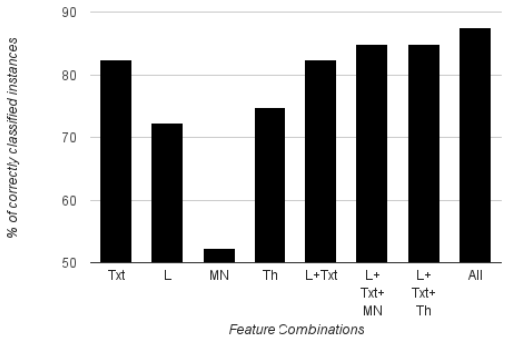


Figure 1. *Percentage of correctly classified instances for different combinations of features*

5. DISCUSSIONS

Most of the misclassifications were observed to be due to lack of information about the importance of certain class of emails or

contacts, that were not represented well enough in the training set of emails. For instance, there were emails from family members or close friends about events like functions, gatherings and meetings which were over-looked as they were small in number and were not captured well due to lack of emails corresponding to similar events in the training data. Also, the importance of activities or events that are related to a person's personal life in terms of his hobbies, interests and so on, were overlooked.

These misclassifications can be resolved by including a top-down perspective to our work. We can get the person to specify important details like hobbies, biographical data, family members, landmarks in personal and professional life, which they feel are of autobiographical value. These could serve as useful cues in identifying emails with autobiographical content that might get misclassified when using the bottom-up approach alone. We could look at a mix of top-down and bottom-up methods in the future to go about gathering emails with autobiographical content and, evaluate how the addition of top-down techniques help in resolving the previous misclassifications and, improve the performance of classifiers.

In our work, we used emails present in the inbox of 3 test users as the primary source of data for our experiments due to a lack of relevant datasets that are publicly available. There are several privacy and content related issues that restrict the availability of emails, and a potential future work would be to come up with a suitable and representative email corpora which can be made publicly available for future use.

Apart from emails, our idea can be extended to other rich sources of personal data like blogs, posts and interactions on social media, all of which have the potential to contain autobiographical content about a person. In the case of emails, we looked at features like labels and filter names, thread counts, one-to-one interactions with contacts in the mail network and so on, in addition to the textual content present in emails. It would also be an interesting problem to identify different social network and temporal properties of interactions in different sources and explore how they can leveraged to gather autobiographical

content for our main goal: generating the life story of a person with minimal supervision.

6. CONCLUSION

In this paper, we have looked at the problem of using online social interactions to create a person's life story. Central to addressing this problem is the ability to discriminate content that is of autobiographical value from the rest. We have identified emails as a rich source of information for the creation of a user's autobiography, conducted preliminary tests and presented the results of our explorations on emails. We observed that textual content and label names present in emails were the most informative individual features and that the combination of textual content, labels, mail network properties and threads gave the best performance, from our classification experiments on a bottom-up approach to autobiography generation. We have also talked about introducing a mix of top-down approaches in the future, to resolve the misclassifications that were observed, and interesting extensions to other sources of personal data like blogs and interactions on social media through posts, etc., to gather autobiographical content for the generation of a person's life story.

REFERENCES

1. Fitzgibbon, A. & Reiter, E. 2004. Memories for life: Managing information over a human lifetime. In T. Hoare & R. Milner (Eds.), *Grand Challenges in Computing Research* (pp. 13-16). Swindon: British Computing Society
2. Ringel, M., Cutrell, E., Dumais, S. & Horvitz, E. 2003. Milestones in time: The value of landmarks in retrieving information from personal stores. To appear in the *proceedings of Interact 2003*.
3. Wachsmuth, H. & Bujna, K. 2011. Back to the roots of genres: Text classification by language function. In proceedings of the *5th IJCNLP* (pp. 632-640).
4. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. & Witten, I. H. 2009. The WEKA data mining software: An update. *SIGKDD Explorations*, 11/1.

5. Ulrich, J., Murray, G. & Carenini, G. 2008. A publicly available annotated corpus for supervised email summarization. In proceedings of the *AAAI EMAIL Workshop* (pp. 77-87).
6. Yang, Y. & Pedersen, J. 1997. A comparative study on feature selection in text categorization. In *ICML 1997* (pp. 412-420).
7. Kiritchenko, S., Matwin, S. & Abu-Hakima, S. 2004. Email classification with temporal features. *Intelligent Information Systems* (pp. 523-533).
8. Cohen, W. 1996. Learning rules that classify e-mail. In proceedings of the *AAAI Spring Symposium on Machine Learning in Information Access*.
9. Manning, C., Raghavan, P. & Schtze, H. 2008. Vector space classification. *Introduction to Information Retrieval*. Cambridge University Press
10. Garera, N. & Yarowsky, D. 2009. Modeling latent biographic attributes in conversational genres. In proceedings of the *Joint Conference of Association of Computational Linguistics and International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, (pp. 710-718).

ESHWAR CHANDRASEKHARAN
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING,
INDIAN INSTITUTE OF TECHNOLOGY MADRAS,
CHENNAI 600036, INDIA.
E-MAIL: <ESHWAR@CSE.IITM.AC.IN>

SUTANU CHAKRABORTI
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING,
INDIAN INSTITUTE OF TECHNOLOGY MADRAS,
CHENNAI 600036, INDIA.
E-MAIL: <SUTANUCG@CSE.IITM.AC.IN>

International Journal of Computational Linguistics and Applications, Vol. 6, No. 2, 2015, pp. 43–59
Received 01/02/2015, Accepted 27/02/2015, Final 07/04/2015.
ISSN 0976-0962, http://ijcla.bahripublications.com

A New Machine Translation Decoder Based on Artificial Immune System

MANEL AMMAR
SALMA JAMOUSSI
Sfax University, Tunisia

ABSTRACT

This paper focuses on decoding as main part of statistical machine translation. Decoding is considering as a NP-complete algorithm that requires intelligent heuristics to and optimum solutions. In order to solve this problem, we proposed a decoder named DAIS based on the meta-heuristic of artificial immune system. The evaluation is performed on two different corpora. The obtained translations show that the proposed approach obtains encouraging results by comparing them to those of the most known decoders in the field.

1. INTRODUCTION

The main task of machine translation (MT) is to translate from one natural language to other target languages. Reduce human intervention is one of the main objectives in this task. A statistical MT (SMT) presents a promising avenue for eliminating experts role, improving translations quality and reducing costs. To make an SMT system, we need three complementary components. First, a language model which is used to and syntactically correct results in the target language. It is trained from monolingual text corpora. Second, the translation model which needs a parallel corpora in order to align the different source sequences with their probable target translations. Then, it assigns a probability value characterizing their appearances in the training corpora. Finally, a decoder produces an optimal statistical translation among all possible translations

by using intelligent search algorithms. These basic components of the SMT can be likened to the noisy-channel approach [1]. Infact, The main goal of an SMT system is to produce the best target sentence *T** for a given source sentence *S*. It is the decoder that will search for this solution in order to compromise between maximizing the translation probability $P(T|S)$ and the language model probability $P(T)$. *T** is found from the following Bayes theorem:

$$T^* = argmax_T P(T|S) = argmax_T P(T) \times P(T|S) \quad (1)$$

In this paper, we focus on developing a decoder called DAIS, abbreviation for Decoding with Artificial Immune System. DAIS is a SMT decoder based on the meta-heuristic of immune systems. DAIS was experimented on a French English translation task and inversely. It uses two fundamental SMT components namely, translation model (TM) and the language model (LM). the obtained results show that our proposed decoder has better performance than Moses decoder. We also scored an acceptable execution time comparable to the time taken by Moses. In this paper, we have studied state-of-the-art decoding task. Next, we present the meta-heuristic of immune system as well as its application in the decoding task in SMT. Finally, we evaluate its performance by comparing the obtained results with DAIS to other named decoder such as Moses and Google translation.

2. RELATED WORKS

The fundamental dilemma of the decoding task is the huge search space. Thus, it is not possible to check all translation possibilities because of the combinatorial explosion of feasible hypotheses. To make the decoding task feasible and effective, the key solution was to use the optimization methods based on intelligent meta-heuristics. These methods are well suited for resolving this type of problems with huge search spaces. We distinguish two broad classes of methods. The first includes local search methods such as beam search, dynamic programming, etc. The second is

the class of global methods as evolutionary methods. Several decoders have been proposed in the literature. The majority of them are based on local optimization methods. This is thanks to their neighbourhood principle allowing the use of a relatively small memory space. We cite in this context, Moses [3] and Marie [4] decoders which realize different implementations of beam search algorithm. We also find in the literature decoders based on dynamic programming [5] and on greedy search [6].

According to our knowledge, the number of decoders implementing bio-inspired methods is very limited. We find only the works of [7] which use genetic algorithm to develop a decoder called GADecoder (Genetic Algorithm Decoder). Since it is a word based decoder, it can find good translations for words but it has difficulties to find the best word ordering. With 4-grams language model, the results were not good enough as Pharaoh [2] and Google Translation. We have noticed that bio-inspired methods are rarely used in SMT field. Indeed, they avoid local optima and provide a global view of the search space thanks to their candidates distributed intelligently and contributed each one to build the final solution.

3. FUNDAMENTALS AND MAIN COMPONENTS OF THE ARTIFICIAL IMMUNE SYSTEM (AIS)

In nature, the immune system plays a very important role in the survival of vertebrates. Indeed, the defense mechanism aims to protect the body against unauthorized intruders no belonging to self named antigens. An immune system is then faced with the problems of detection, identification and response to pathogens. The first step of an immune system is to determine antigens no belonging to self by the mechanism of negative selection. Then clonal selection mechanism will be trigged in order to prepare the adequate immunity response allowing their elimination. The basic actors in the immune system are the white blood cells or lymphocytes. They produce receptors called antibodies, with which they can discriminate self from non-self and neutralize pathogens elements. Neutralization of an antigen is made by a connection between the antibody and a specific part of the

intruder antigen. More the degree of affinity of this connection is strong, the system is able to eliminate the concerned antigen. There are two main types of natural immunity. The first one is the innate immunity; it is an elementary response against an antigen firstly encountered in the human body. The second is adaptive immunity; it is a stored immune response indicating that the human body has already identified this antigen and the antibody specific for its neutralization is already built and ready to respond.

In computer science, the artificial immune system (AIS) is an interesting biological inspiration for solving optimization problems [8]. On a computer view, the negative selection algorithm is based on the idea of generating a set of malicious elements and tests the ability of such system to determine them [9]. Thus, the clonal selection algorithm uses the set generated by the negative selection algorithm in order to nd the best solution to the problem. It aims to produce an initial population of solutions. These solutions will be cloned several times according to their degrees of affinity with the antigen. Clones results will undergo mutations by modifying some of their characteristics in order to attain a higher degree of affinity with the antigen. Algorithm 1 explains the general principle of artificial clonal selection.

```
Algorithm 1: The artificial clonal selection
Input: S= a set of antigens to be recognized
Output: M= a set of best antibodies met\\
  Begin
  Create an initial random set of antibodies, A
  for each Si in S
    Determine the affinity of Si with each antibody in A,
    Generating clones for each antibody in A
    proportionally to its affinity degrees
    Mutate attributes of these clones
    Place a copy of the highest affinity antibodies into
    the memory set, M
```

Figure 1. *The principle of the clonal selection algorithm for the SMT decoding problem*

4. AN ARTIFICIAL IMMUNE SYSTEM FOR MACHINE TRANSLATION

AIS meta-heuristic has a number of features that are so interesting to test them for resolving the decoding problem in SMT. Indeed, there are common points between the decoding as a computer problem and the functioning of a real immune system. The latter one must react quickly and correctly to protect the body. For an effective immune response, the human body must have the necessary information on the specifics of each antigen. The same for the decoding problem in SMT, the goal is to acquire characteristics of pair of language in order to produce quickly and efficiently target translations. In addition, the clonal selection technique enables antibodies to multiply proportionally to their quality of neutralization of an antigen; we notice that its implementation will promote the less costly translations. To apply AIS for the SMT task we need first to define some preliminary concepts:

- The antigen in SMT (*AG*): it is the source sentence to be translated expressed in a source language;
- The antibody (*AC*): the intended aim; it is the element allowing neutralization of the antigen. It corresponds to the target sentence expressed in the desired language. A good antibody must be both faithful to the source sentence content and syntactically correct in the target language;
- The innate response is the policy that we have adopted for solving the decoding problem. We considered each source sentence as a new antigen to be neutralized.

5. THE NEGATIVE SELECTION ALGORITHM

The negative selection algorithm enables the immune system to discriminate self from non-self cells. Transposing this principle to our decoding task seems very useful. Infact, upon penetration of an antigen that corresponds to a non-self cell, the artificial negative selection is triggered. First, it locates the area in the

search space in which we can find the target solution. So, it generates a set of translation options denoted *D*. This set is composed of translation options which each option is defined by a source fragment, a target fragment, the *TM* value and the number of source words covered by this option. Second, we prepare and reduce the set *D* in order to promote the most promising translation options. We eliminated unattractive options using the pruning technique. It consists in eliminating all translation options having high cost, and this by comparing the TM value of each option to a fixed threshold. The following algorithm 2 describes the mechanism of negative selection adopted in our decoding algorithm.

```
Algorithm 2: Negative selection algorithm in DAIS decoder
Input E= the search space composed of translation options: ei
while (i <Card(E))
  if(ei covers a part of AG )
    if(TM(ei) < thresholdPruning )
      ignore the option ei from the search space
    else
      sort ei in set D taking into account the number of words
      covered in the source sentence and its TM value
      (equiprobable scheduling
```

Figure 2. *Construction of the search space for a given antigen AG*

6. CLONAL SELECTION ALGORITHM

The clonal selection aims to produce specific antibodies allowing the neutralization of pathogens. In our case, the clonal selection algorithm is the heart of our translation system. Indeed, we admit that the artificial neutralization consists in finding the best target translation of a given source sentence. The principle of this algorithm illustrated in Figure 3 is based on four operators namely evaluation operators, cloning, mutation and communication. The initialization of our translation system consists in producing a random set of antibodies population. So, we need to use the subspace given by the negative selection algorithm to form a set of initial hypotheses in the target

language that we called initial antibodies. The system thus, needs to improve those hypotheses to find the best antibody.

6.1. *Affinity evaluation*

To evaluate the affinity of selected antibodies, we adopted the same scoring function implemented in Moses decoder [3]. The goal is to find the antibody T (target sentence) for a given antigen S (source sentence), which maximizes $Affinity\ (T, S)$. This score function is presented as follows:

$$Affinity\ (T, S) = \lambda_{tr} \times log(TM(S|T)) + \lambda_{lm} \times log(LM\ (T)) - \lambda_{w} \times log(exp^{w}\ (T)) \quad (2)$$

Where $TM(S|T)$, LM(T) and $exp^{w\,(T)}$ represent respectively the translation model probability, language model probability and the penalty on the length of the target sentence T. λ_{tr}, λ_{lm} and λ_{w} are the weights of the translation model, language model and length penalty. Indeed, a good antibody is one that maximizes the objective function Affinity (T, S). To estimate lambda, we use MERT tuned system from Moses decoder.

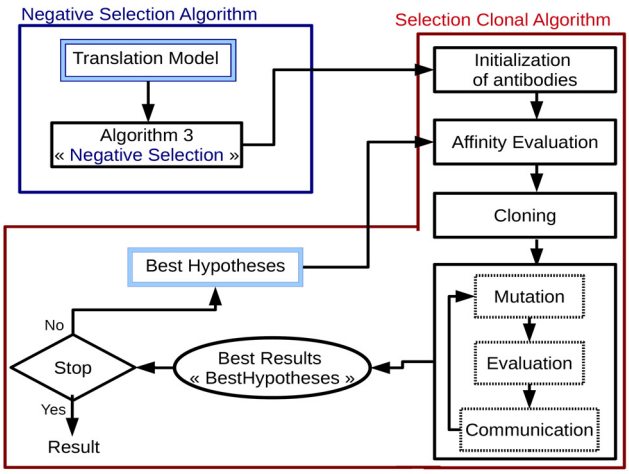


Figure 3. *The principle of the clonal selection algorithm for the SMT decoding problem*

6.2. *Cloning*

The cloning technique is to make identical copies to hypotheses proportionally to their evaluation scores. As a result, the antibody that has the best affinity score has the higher number of clones. The following formula calculates a percentage characterizing number of clones that deserves a given antibody:

$$nbClones_{AG}(AC_i) = \frac{Affinity(AC_i, AG)}{\sum_{j=0}^{NCI} Affinity(AC_j, AG)} \times NCI \quad (3)$$

Where NCI is the number of initial candidates. This parameter had been set empirically.

6.3. *Mutation*

The mutation alters the characteristics of different clones in order to evolve them. We implemented two types of mutation. The first one substitutes an option in an antibody by another from the search space. The second consists in changing the positions of translation options in a given antibody.

The mutation of translation options: In a first step and for each clone we fixed the number of mutations ($NM(AC_i)$) to the number of translation options presented in AC_i. Each iteration of mutation begins by randomly choosing the translation option to be mutated (OP_j). Then, we specify the number of mutations that merit OP_j. Thus, a translation option OP_j having a low a affinity requires more mutation than another option having improved a affinity. The following formula seeks the number of mutations required for a translation option $nbMut(OP_j)$ in an antibody AC_i:

$$nbMut(OP_j) = (1 - \frac{TM(OP_j)}{\sum_{k=0}^{NM(AC_i)} TM(OP_k) + \varepsilon}) \times NM(AC_i) \quad (4)$$

The second step consists in consulting the search space to select the new translation option that will replace the old one. So, we used the method of wheel selection explained in [7]. Indeed, this

algorithm is applicable for ordered search space that corresponds well to our case.

The mutation of positions of translation options: Translation from source into target language requires often changing positions of words in the target sentence to be syntactically correct in the target language. In our decoder, a translation option can interchanges its position only with options that directly precede or follow it. A correct inversion of position words is useful information to communicate to other antibodies. This communication is elaborated through an enrichment of the search space by a new translation option in order to take advantage for the next clones.

At antibody level, a good mutation signifies that the latter was able to improve it's affinity score. So it can be very close to neutralize the source sentence. For this, we gave him a chance to make another mutation ($NM(\mathrm{AC}_i) = NM(AC_i) + 1$). On the contrary, an antibody which makes a deteriorating mutation of it's a affinity score will be penalized by reducing its chance of one mutation ($NM(AC_i) = NM(ACi) - 1$).

6.4. *Communication between clones*

For bio-inspired algorithms, communication between candidates represents the ideal solution to cope with the combinatorial explosion and improves the convergence speed and quality of AIS algorithm.

Communication adopted by the AIS meta-heuristic: According to Figure 3, the AIS algorithm uses antibodies results of iteration i to establish the iteration $i + 1$. These antibodies are not only the initial candidates for the new iteration, but also a new search space for this iteration. Thus, to improve the quality of solutions, two possibilities can be envisaged: either we increase the number of antibodies participating in the search for a solution. As a result, we obtain a rich search space in each iteration. Or we try to improve the quality of antibodies in iteration i, so the algorithm in this case, requires only a few iterations to converge. Due to limited capacities of our machines,

we chose the second possibility and this by proposing a new type of communication for AIS algorithms.

Proposed communication: The proposed technique consists in taking advantage of the information provided by an antibody when it mutes one of its translation options or change its positions. The affinity score allows us to measure the impact of this mutation on the translation quality. Thus, if the score has improved after a replacement of an option OP_{anc} by other OP_{new}, we propose then, to record a slight improvement (bonus) in the TM value of OP_{new} using the following formula:

$$TM_{OP_{new}} = TM_{OP_{new}} + (TM_{OP_{new}} \ bonus) \quad (5)$$

Conversely, if an option does not improve the quality of the translation, then it undergoes a reduction to its TM value ($malus$) as follows:

$$TM_{OP_{new}} = TM_{OP_{new}} - (TM_{OP_{new}} \times malus) \quad (6)$$

Note that the couple ($bonus$, $malus$) is determined empirically. So, clones have right to change the structure of the search space (fostering option over another).

6.5. *Stop condition*

The evaluation process, cloning and mutation are repeated until a finite number of iterations. In natural immune system, one mutation operation creates a new antibody different from original which in turn is susceptible to undergo a new cloning iteration. The application of AIS algorithm in this way will cause an exponentially higher complexity. This is caused by the cloning operator allowing an exponential increase in the number of candidates. To avoid this problem and make possible the application of AIS algorithm, we conducted a series of mutations on each antibody before starting a new cloning iteration (Figure 3). The number of iterations taken by DAIS to converge to a good solution is estimated as follow:

$$NbItr\ (AC_i) \simeq NbClones(AC_i)\ \text{X NM(ACi)}\ \text{X cts} \quad (7)$$

We can conclude that $nbItr$ is a dynamic parameter that is influenced by the length of the source sentence, the number of initial candidates (NCI) and the constant (cts is a constant indicating the number of times that cloning process was repeated). A good configuration can give better results in a reasonable time. For example, for a short source sentence, we have set a very low cts to avoid a long and unnecessary calculation. For a long source sentence, it would be effective to increase the number of antibodies NCI and cts to explore more feasible solutions before converging.

The general algorithm of the DAIS decoder is illustrated in algorithm 3.

7. Results and Discussion

7.1. *Corpora*

An SMT process requires two basic steps: learning and testing. Each one requires bilingual and aligned corpora. Both training and test corpora must be from the same domain. DAIS decoder has been tested on two different corpora: French and English sentences. First, we used the English French "WIT[3]" [10] parallel corpora extracted from TED Talks. Second, we test our translation system on a French English bilingual corpora that is "OpenOffice[1]" containing a collection of documents describing the functions offered by the open office tools. Table 1 summarizes the characteristics of these two corpora. In the training phase, the translation model is obtained using the GIZA++ tool [11]. It allows the alignment of bilingual corpora in order to extract sub-translations with their probabilities. The language model is learned respectively on the French and English monolingual corpora. It is used to assign a probability value to a word sequence to qualify the correct construction of this latter in

[1] http://opus.lingfil.uu.se/OpenOffice.php

the target language. DAIS decoder uses a tri-gram language model generated by SRILM tool [12].

7.2. *Evaluation of DAIS decoder*

The adopted translation system requires the adjustment of a number of parameters. So, we make several attempts to detect the best configuration. The best parameter values found are summarized in Table 2.

```
 Algorithm 3: General algorithm for DAIS decoder
Input: AG = antigen (sentence to be translated)
--Application of negative selection algorithm--
  construction of search space of AG according to the algorithm 2
--Application of clonal selection algorithm--
  initialization of NCI candidates
  repeat
    while(i <NCI)
    Evaluation: score= Affinity(ACi, AG) (Equation 2)
    Cloning: nbAC= clonage(ACi) (Equation 3)
    while(j <nbAC)
      NM(ACij) = Card(ACij)= number of translation options in ACij
      while(k <NM(ACij))
      Mutation: Randomly select an option to be mutated: OP
      nbOP: number of mutations that deserves OP (Equation 4)
      while (r <nbOP)
         Substitute OP with another from the search space using
         wheel selection algorithm
         Evaluation: score' after this mutation
         if(score' > score)
            Assign a bonus to OP (Equation 5)
            Another chance to mutation given for (ACij)
        else
            Mutate the position of OP with its neighbors: OPneigh
            Evaluation: score'' after this mutation of position
            If (score'' > score')
                Add the new OPneigh to the search space
            else
              Assign a malus to OP (Equation 6)
              Reduce the number of mutation given for (ACij)
    until(cts): number of iterations was satisfied.
 return ACmax: the best antibody met
```

Figure 4. *Algorithm 3: General algorithm for DAIS decoder*

The evaluation is based on two main criteria, the quality of the translation reflected by score BLEU [13] and score TER [14], and the execution time. We compared our DAIS decoder with other reference decoders such as Moses and Google translator. Table 3 shows results obtained with the test corpora "test2012" by the different translators (DAIS, Moses and Google translation). We can notice that the BLEU values obtained by DAIS are better than those obtained by Moses. For example, for 4-grams, our DAIS decoder gets a BLEU value equals to 19:29 against 13:71 obtained with Moses decoder. Results also showed that Google translator generates the better quality translations than DAIS and Moses with a BLEU score equal to 42:12.

For the TER score, we found 0:65 translation error rate for DAIS decoder in test2012. We are better than Moses which has 0:68. Google is the best with TER equal to 0:46. Note that TER is an error metric for machine translation that measures the number of edits required to change a system output into one of the references.

Table 1. *Characteristics of the used corpora*

Corpora	WIT[3]		OpenOffice	
	English-French		French-English	
Type	Train	Test2012	Train	Test
Number of sentences	186510	1124	24500	1000

Table 2. *The best parameter values found after several execution attempts*

Parameters	**test2012**	**OpenOffice**
nbItr	1000	200
(λ_{tr} , λ_{lm}, λ_{w})	(3 , 2 , 0.7)	(1 , 2 , -1)
NCI	50	20
(bonus, malus)	(0.1 , 0)	(0.1 , 0)

In Table 3, we note that DAIS required an execution time higher than Moses decoder (an average of 1 minutes per sentence with DAIS against 0:5 minutes with Moses). Google has the most efficient time translation with a mean of 0:02 seconds per sentence.

Indeed, we share the same learning base with Moses (the translation model and the language model). However, Google translator uses an efficient language model that covers almost all the linguistic peculiarities of the French language and a complete translation model.

Also, to ensure that DAIS decoder is independent of the pair of used languages, we test it with a French English corpora which is "Open Office". We used a sample of 1000 French sentences sources to compare the behavior of the different translators. From Table 4, we note that the results clearly confirm our previous conclusions. The obtained results are excellent even exceeding those given by Google with a gap equal at 3:4. Furthermore, we note that our DAIS decoder leaving an acceptable execution time equal to 12 seconds per sentence. However, it is higher than the time taken by Moses (3 seconds per sentence) Google (0:002 seconds per sentence). Same for the TER score, DAIS is the best with TER = 0:46, the second is google TER = 0:69 and the third is Moses TER = 0:89. We note that the obtained results with OpenOffice corpora are better than those obtained with WIT[3]. This implies that our decoder performs better from French to English than vice versa.

Table 3. *Comparison of translation results in terms of BLEU score and execution time*

nGram	**Google**	**Moses**	**DAIS**
4-grams	42.12	13.71	19.29
Time: min/ph	0.02	0.5	1

Table 4. *Comparison of translation results in terms of BLEU score and execution time obtained by DAIS and Google with the test corpora of "Open Office"*

Comparison characteristics	**Moses**	**Google**	**DAIS**
4-grams	10.72	22.4	25.8
Execution Time: sec/ph	3	0.002	12

8. CONCLUSIONS AND PERSPECTIVES

The decoding is a main part of statistical machine translation. It aims to find the right combinations of target fragments by using a decoding algorithm. The optimization method must manage a difficult compromise between the quality of translations and the execution time. In this context, we used the artificial immune system algorithm inspired by the biological immune system. Indeed, we have developed a SMT decoder named DAIS that enables each candidate to build its own solution and communicate with others candidates in order to form cooperatively the best solution for the problem. We trained and tested our decoder system on two corpora; "WIT3" and "Open Office". To evaluate the performance of DAIS, we compared it to other known decoders as Moses, which is based on beam search algorithm and the Google translator that uses very large databases. The comparison is based on translation quality measured by BLEU score and execution time. We have noticed that DAIS decoder gives better performance than Moses. Results are generally quite encouraging, they are comparable to those obtained by Moses but they are still further than those obtained by the Google translator thanks to the richness of its database.

In future work we will focus on the concept adaptive reaction. It seems more efficient and especially when we increase the analysis level of the source sentence taking into account its syntactic or semantic characteristics. Also, we think to parallelize our decoder. Indeed we use a bio-inspired algorithm that demands large computing time and memory. In SMT the most expensive step is the performance evaluation of hypotheses. This calculation is totally independent of one to another hypothesis. Whence it seems profitable to parallelize our decoder to improve quality and reduce costs.

REFERENCES

1. Brown, P., Della Pietra, V., Della Pietra, S. & Mercer, R. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19, 263-311.

2. Koehn, P. 2004. Pharaoh: A beam search decoder for phrase-based statistical machine translation models. In the *6th Conference of the Association for Machine Translation in the Americas* (pp. 115-124), *AMTA '04*, Washington, DC, USA.
3. Koehn, P., Hoang, H., Birch, A., Callison-burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A. & Herbst, E. Moses. 2007. Open source toolkit for statistical machine translation. In proceedings of the *45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, ACL '07* (pp. 177-180), Association for Computational Linguistics Prague, Czech Republic.
4. Crego, J. & Marino, J. 2007. Extending MARIE: an N-gram-based SMT decoder. In *Meeting of the Association for Computational Linguistics*. *ACL '07* (pp. 213-216), Prague, Czech Republic.
5. Niessen, S., Vogel, S., Ney, H. & Tillmann, C. 1998. A DP based search algorithm for statistical machine translation. In *Meeting of the Association for Computational Linguistics* (pp. 960-967), *ACL '98*, Montreal, Quebec, Canada.
6. Germann, U., Jahr, M., Knight, K., Marcu, D. & Yamada, K. 2001. Fast decoding and optimal decoding for machine translation. In *Meeting of the Association for Computational Linguistics* (pp. 228-235), *ACL '01*, Toulouse, France.
7. Ebadat, A., Smaili, K. & Langlois, D. 2009. Using Genetic Algorithm for the Decoder. Master's thesis, Saarland University, Computational Linguistics & Phonetics.
8. Watkins, A., Timmis, J. & Boggess, L. 2004. Artificial immune recognition system (airs): An immune inspired supervised learning algorithm. *Journal of Genetic Programming and Evolvable Machines* (pp. 291-317).
9. Abdelhadi, A., Mouss, H. & Kadri, O. 2010. Algorithmes du syst↕ eme immunitaire artificiel pour la surveillance industrielle. In *International Conference on Industrial Engineering and Manufacturing ICIEM'10*, Batna, Algerie.
10. Cettolo, M., Girardi, C. & Federico, M. 2012. Wit3: Web inventory of transcribed and translated talks. In proceedings of the *16th Conference of the European Association for Machine Translation (EAMT)* (pp. 261-268), Trento, Italy.
11. Och, F. & Ney, H. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29, 19-51.
12. Stolcke, A. 2002. Srilm-an extensible language modeling toolkit. In: proceedings of the *7th International Conference on Spoken*

Language Processing (pp. 901-904), *ICSLP '02*, Denver, Colorado, US.
13. Papineni, K., Roukos, S., Ward, T., j. & Zhu, W. 2002 Bleu: A method for automatic evaluation of machine translation. In *Meeting of the Association for Computational Linguistics* (pp. 311-318), *ACL '02*, Philadelphia, PA, USA.
14. Snover, M., Dorr, B., Schwartz, R., Micciulla, L. & Makhoul, J. 2006. A study of translation edit rate with targeted human annotation. In proceedings of *Association for Machine Translation in the Americas* (pp. 223-231).

MANEL AMMAR
INSTITUT SUPÉRIEUR D'INFORMATIQUE
ET DE MULTIMÉDIA DE SFAX,
SFAX UNIVERSITY,
SAKIET EZZIT, SFAX, TUNISIA.
E-MAIL: <MANEL1401@GMAIL.COM>

SALMA JAMOUSSI
INSTITUT SUPÉRIEUR D'INFORMATIQUE
ET DE MULTIMÉDIA DE SFAX,
SFAX UNIVERSITY,
SAKIET EZZIT, SFAX, TUNISIA
E-MAIL: <JAMOUSSI@GMAIL.COM>

International Journal of Computational Linguistics and Applications, Vol. 6, No. 2, 2015, pp. 61–82
Received 23/01/2015, Accepted 27/02/2015, Final 06/04/2015.
ISSN 0976-0962, http://ijcla.bahripublications.com

knoWitiary: A Machine Readable Incarnation of Wiktionary

VIVI NASTASE
CARLO STRAPPARAVA
FBK-irst, Trento, Italy

ABSTRACT

knoWitiary *is a resource that presents a reorganized version of Wiktionary's information in machine readable format. Wiktionary contains a plethora of information about words, including sense definitions, etymology, translations, derived terms and anagrams. Similar work to the one reported here goes one step further than extracting information from Wiktionary: mapping it onto WordNet – NLP community's de facto gold standard. Lexical and relation overlap shows that Wiktionary provides different types of information compared to WordNet, which implies that much is discarded when doing a mapping. We make a case here for making space for "pure" resources alongside mapped ones, to preserve the unique information that idiosyncratic resources such as Wiktionary provide, which may open up new avenues to explore for tasks that require varied and "unorthodox" information about words.*

1. INTRODUCTION

knoWitiary is a network of words and senses obtained exclusively from Wiktionary. We compare it with one of the most used resources in NLP – WordNet. Because WordNet has been in use for more than two decades now, we know its strengths and weaknesses. Work on resources that are similar to it – deal with words, their senses and relations between them –

often falls back onto WordNet, to take advantage of its manually built (and therefore, we consider it) gold standard hierarchy and inventory. By "fall back" we mean that instead of building a new resource from scratch, the work concentrates on adding to this resource, enriching it with new words, senses or relations between them. This may seem appropriate from several points of view – the "backbone" of the newly proposed resource is not called into question since it is WordNet itself; adoption of the new resource can be faster, since it adheres to WordNet's format and conventions. There may not be only advantages in adopting WordNet as the core of a new resource. First, criticisms of WordNet itself – its occasionally too fine-grained sense distinctions, differences in the level of detail in hierarchies governed by different "top senses" – would apply to the new resources. But the most important disadvantage of using WordNet as a scaffolding to build upon is that the structure and inventory it imposes enforces compromises in the kind of knowledge that is being added to it: mapping a dictionary or Wikipedia onto WordNet requires a mapping at the level of senses, and there does not exist a one-to-one mapping between entries in these resources and WordNet. This leads to further cuts in the amount and type of information that was extracted and could have been made available. Thus, the mapping causes a compromise, with loss of information and structure. These losses are not quantified in such work, as the more positive aspects – the mapping process and the enrichment of the "base" resource, usually WordNet – are emphasized (e.g. [1, 2]).

We proceed to build a stand-alone resource by formalizing Wiktionary as comprehensively as possible. Wiktionary[1] is a very rich dictionary, that presents in semi-structured format a plethora of information about words: senses and glosses, phonology, derivations, word relations within a language and across languages. It also contains "unorthodox" relations, such as ANAGRAMS[2] and ETYMOLOGY. This treasure trove of

[1] http://en.wiktionary.org/wiki

[2] Incidentally, note that *knoWitiary* is an anagram of *Wiktionary*

interconnected information is unprecedented, and could bring new exploration avenues for established NLP tasks, and the needed spark for novel creative language tasks. We compare this resource with WordNet – the English 3.1[3] and the Italian versions[4]. The comparison shows a large amount of novel information, only part of which can be imported when performing a mapping [3].

Versions of Wiktionary formatted for machine consumption already exist, including a freely available Java library for processing the Wiktionary dump (JWKTL). Each of these has some piece missing. We will review these versions in Section 2. In Section 3 we describe knoWitiary and its general statistics in terms of multi-lingual lexica and relations. The comparison with English and Italian wordnets is described in Section 4, and we wrap up with a brief overview of tasks that could be aided or made possible by having a resource with the kind of varied information that Wiktionary contains in Section 5.

2. RELATED WORK

WordNet [4] has for many years been the lexical resource used in NLP research. Built by psycholinguists and lexicographers, its structure relies on the notion of synset – a set of one or more synonyms that expresses a "unit" of meaning, linked through several types of lexico-semantic relations with other synsets: semantic (e.g. HYPERNYM, HYPONYM; three types of MERONYM and HOLONYM; SYNONYM, ANTONYM); lexical (DERIVED FROM, PERTAINYMS, PARTICIPLE OF); domain / member of domain. An index serves to map word forms under four parts of speech (adjective, adverb, noun, verb) onto synsets, and data files include the relations between synsets.

WordNet has provided the gold standard in word senses (used as reference for multiple word sense disambiguation exercises within Sens-/Sem-Eval) and ontology (used as a

[3] http://wordnet.princeton.edu/wordnet/download/
[4] http://multiwordnet.fbk.eu/english/home.php

reference for ontological relation extractions) for the general English language. Having such a strong backbone, rather than build something new from scratch, there have been efforts to produce enhanced wordnets – with coarser word senses [5], in different languages [6], with sentiment annotation [7, 8], with domains [9] with more relations [10-12].

Wiktionary[5] is an online collaborative dictionary, companion to **Wikipedia**[6], which provides a collaborative wiki platform for the building of dictionaries in multiple languages. Reflecting the varied knowledge of the contributors, Wiktionary contains comprehensive information about words. For a given *word form* we may find each applicable part of speech and language, alternative spellings, (layered) senses and definitions, phonology, etymology, synonyms, hypernyms, hyponyms, derived and related terms, translations, anagrams, and images.

Just like Wikipedia, Wiktionary is semi-structured: it has sections for each of the main types of information it provides, while the information within the section can be structured or not. For *derived forms* for example we find the related entries listed, while etymological information appears in a free-form paragraph, but which contains structured word information, and often regular patterns to express etymological links.

Various types of information extracted from Wiktionary have been exploited successfully for a variety of NLP tasks, such as semantic relatedness measures [13], cross-language image retrieval [14], named entity recognition [15], synonymy mining [16], cross-language text categorization [17]. Such results have shown that Wiktionary is a desirable resource, and its availability in machine-readable format would be an asset to NLP applications.

JWKTL[7] is an API to Wiktionary, available as a free Java library. It processes a Wiktionary dump and populates a database with information for English, Russian and German. The library

[5] http://en.wiktionary.org
[6] http://en.wikipedia.org
[7] https://www.ukp.tu-darmstadt.de/software/jwktl/

provides very fast processing of the Wiktionary dump, and varied and flexible methods to access the formalized information. Because the entries in the dictionary are not all structured, some information is lost in conversion. In particular, the etymological information is presented as a string (the paragraph as it was on the wiki page), without further formalization. With respect to the other information contained therein, and bar some parsing errors on either side, this resource and knoWitiary and roughly equivalent.

de Melo [18] describes *Etymological WordNet*, built from etymological links mined from the *Etymology* and *Derived from* sections, and also definitions. This resource is part of a larger repository, described in [19], built based on a few assumptions that define and prescribe the definition and design of universal multilingual knowledge bases. The concrete work done towards achieving such a comprehensive resource uses WordNet as the base, and it builds upon it by adding mono-lingual (in particular language family information under the corresponding synset for *language*) and multi-lingual entries (based on translations from Wiktionary), and novel – etymological – links. At the time of writing, this resource was not available for analysis and comparison.

Mappings Since WordNet, Wiktionary and Wikipedia do not subsume one another, there has been effort in various combinations of mapping between the three [2]. Any such mapping imposes the adoption of one resource as the "base", onto which the others are mapped. Because of its good reputation and ubiquity in the field, this "base" is (usually) WordNet.

Miller and Gurevych [2], Gurevych et al. [20] present mappings between WordNet, Wiktionary, Wikipedia and other sources, and include overviews of previous work on mapping between various combinations of such resources. One interesting thing to note is that while considerable effort is made to make, evaluate and report the mapping at the level of nodes, it is not clear what happens with the relations from other sources, or with the un-mapped nodes. Through the mapping, the aim is to show how much the resource (onto which the mapping is done) gets

enriched, but we are missing the final picture – what does the final resource contain. By not investigating the un-mapped portions, we don't know how much of the potential of the other resources remains untapped.

3. FROM WIKTIONARY TO KNOWITIARY

The entries in Wiktionary are semi-structured. There are sections for definitions, etymology, pronunciation, and for each part of speech that may apply, there are derived terms and translations. Different etymologies for the same word are presented in different sections – thus allowing, if necessary, the distinction between homonymy and polysemy. In terms of word senses, there are both coarse and fine-grained distinctions, where a coarse sense may have several sub-senses. This type of information is illustrated through the entry for the word *form* in Figure 1. As apparent from the figure, we note that Wiktionary is organized by word forms. If the same word form appears in another language than English, it appears within the same Wiktionary page, with the same type of information as for English.[8]

[8] We remind the reader that we are processing here only the English Wiktionary (the version from 10.04.2014) where all information apart from the words themselves (if needed) is given in English. It covers however word forms in multiple languages. Wiktionaries for other languages exist, but were not included in the resource described here.

Figure 1. form *in Wiktionary*

We process the structured portions of the page, and for each section of interest extract the available information. We extract first the words and their possible senses and sub-senses with the associated definitions (Figure 2). 29 of the languages represented have each more than 10,000 entries, under 16 parts of speech. Table 1 shows part of the lexicon and relation statistics, for some of the most populated languages.

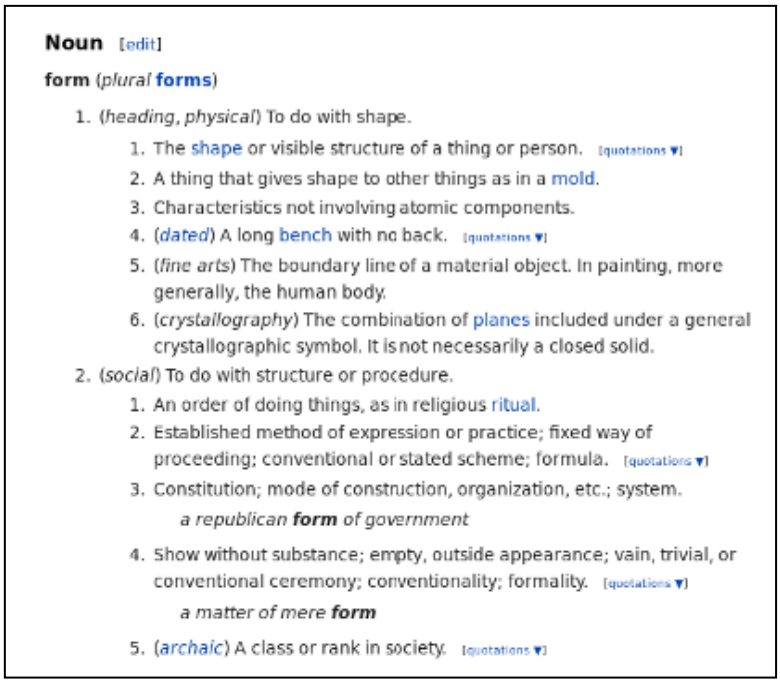


Figure 2. *Definitions*

Table 1. *Selected lexicon and relation statistics in knoWitiary*

Language	# entries	# senses	# subsenses	# relations
English	581,586	702,575	706,466	710,396
French	291,291	126,142	126,202	84,181
German	169,118	231,692	231,727	307,872
Italian	529,630	270,341	270,394	671,689
Latin	661,642	634,588	635,982	589,674
29 most frequent	3,605,984	3,610,320	3,616,321	3,307,981

For each form there is varied information, including related terms, synonyms, antonyms, derived terms, as illustrated in Figure 3. An overview of the relations extracted from these sections (with statistics covering the 29 most represented languages) is included in the top part of Table 2.

Figure 3. *Related words*

Table 2. *Relation statistics in knoWitiary*

General relations		
Relation	**freq.**	**Example**
ACRONYM OF	572	NATO / North Atlantic Treaty Organization
ALTERNATIVE FORMS	91,781	encyclopedia / encyclopaedia
ANAGRAMS	442,422	dictionary / indicatory
ANTONYMS	47,090	free / bound
COMPOUNDS	16,969	live (adj) / live broadcast
CONJUGATION OF	991,759	it:abbreviate / it:abbreviare
DERIVED TERMS	305,339	book (noun) / bookworm
DESCENDANTS	44,069	la:dictionarium (noun) / en:dictionary
HOLONYMS	856	nucleotide / dezoxyribonucleic acid
HYPERNYMS	46,905	mouse / rodent
HYPONYMS	46,908	deer / buck
MERONYMS	856	conjunction / conjunct
RELATED	550,731	lexicography / lexicon
SEE ALSO	106,146	dictionary / vocabulary
SYNONYMS	360,779	book (noun) / tome
total	3,053,182	
Etymological relations (direct)		
Relation	**freq.**	**Example**
ABBRV	365	en:bot / en:robot
BORROWING	2,782	fr:sandwich / en:sandwich
COGNATE	4	en:meal / nl:moal
COGNATE COMPOUND	5	nl:Aalderik / goh:adal:noble + goh:rihhi:ruler
COMPOUND	54,685	la:dictionarius / la:dictio:speaking: + la:-arium:room, place:
CONFIX	8,504	en:morphology / en:morpho
ETYM	188,448	en:dictionary / la:dictionarium
ETYMTWIN	6	en:word / en:verb
total	254,799	

The etymology of words is presented in a "free form" paragraph, which however uses a rather consistent lexicon and expressions to present the information. Figure 4 shows an example. To extract the etymological chain, we parse the *Etymology* section using regular expressions. Statistics on the direct links extracted from these sections are shown in the second half of Table 2.

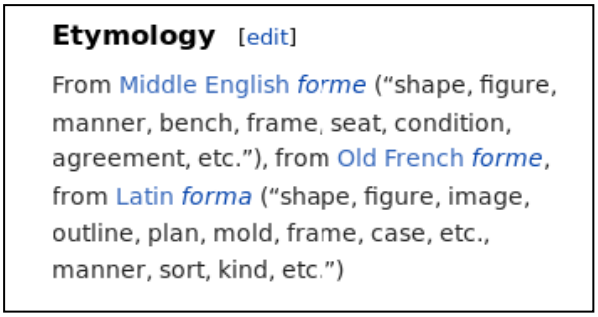


Figure 4. *Etymology in Wiktionary*

Because the Wiktionary entries are "stand alone" (edited individually, and not necessarily coordinated with existing entries), during the construction process we add the symmetrical (ANTONYM, SYNONYM, RELATED) and opposite relations (MERONYM/HOLONYM, HYPERNYM/HYPONYM) to those explicitly given. From the translation section we extract translations. Translations are grouped by sense, and as a link to the word senses a brief version of the sense definition is given, as can be seen in Figure 5.

Figure 5. *Translations by sense*

There are 122,571 entries for 74,384 English words, which have 1,494,527translations. There are multiple entries per word because the translations are at the sense level. There are 2,384 languages represented, 36 of which have more than 10,000 occurrences as translations.

It can be argued that Wiktionary is not a trustworthy source of lexical knowledge, because of its open nature and its collaborative and (probably) non-expert pedigree. We discuss below the issues of lexical material. That Wiktionary contains useful information is evidenced by its contribution to NLP tasks, as explained in Section 2.

Lexical material: root forms There are entries in Wiktionary that do not appear in dictionaries such as Merriam-Webster, Collins, Oxford – e.g. widespreadness, tiltability, hypotheticality. Professionally built dictionaries go through a rigorous process of selecting the lexical material. Word usage is surveyed through selected sources of text, and novel words are

"quarantined" until they become established in the language[9]. This "quarantine" was on the order of years, but has become shorter to keep up with the productivity of language speakers and the high rate of information exchange and spread facilitated by the web and numerous electronic social platforms. The three words mentioned above do not (as yet) qualify for inclusion in such dictionaries, but each has thousands of hits on the web, and appear in various sources including scientific or technical publications. Having an up-to-date inventory of language, even if some entries are destined to fall by the wayside, is useful, whether dealing with contemporary texts, or dealing with older texts that contain words that in the meantime have disappeared from language. There may be situations when Wiktionary contributors may add a new, made-up word that they like. It is not likely that entries like this affect the rest of the resource: if a word that is included is never used, it is not an issue.

Lexical material: inflected forms For inflective languages, such as Italian, the inflected forms may appear as separate entries, whereas proper dictionaries include only the root form and the applicable inflectional rules. These decisions were necessary for paper dictionaries for reasons of space. In an electronic version it is not necessary to censor inflected forms. The statistics in Table 1 & 2 show that the Italian entries cover a high number of inflected forms. We argue that this is neither a problem, nor a negative aspect of the resource. From a practical point of view, inflected entries in the machine readable version of the resource makes recognition easier and faster by simple string match, without need for lemmatization or stemming. There is also another aspect related to the inflected forms, which explains why they are included in the Wiktionary at all: as mentioned before, Wiktionary is organized by word forms. The same word form may appear in different languages, whether as a root or inflected form. All but 16,200 of the forms that have an entry in Italian have entries in other languages as well. For example, the

[9] http://www.oxforddictionaries.com/words/how-do-you-decide-whether-a-new-word-should-be-included-in-an-oxford-dictionary

inflected form *minute* – the plural feminine version of the adjective *minuto* (tiny) – has entries in English, French and Latin as well, some which are inflections (for Italian and Latin) some of which are root forms (for English and French). This parallel between forms in different language is itself an interesting bit of information which is captured by the resource.

4. COMPARISON WITH WORDNET

Since WordNet is the most commonly used ontology in NLP, the question of how the new resource compares comes naturally. For the English and Italian portions of the extracted resource we perform comparison with the corresponding wordnets in terms of lexicon – entries and senses – and relations.

The purpose of the comparison is to quantify both the portions that can be mapped, but most interestingly, those that cannot. The fact that much information cannot be mapped supports the idea that on the field of lexical resources there should be space for "pure" resources other than those manually built by experts, and onto which mappings are done.

4.1. *Lexical comparison*

Table 3 provides numbers for comparison in terms of senses between WordNet and knoWitiary. An apparent advantage of working with Wiktionary is the fact that it has a two-level structure – senses and subsenses, which would allow access at varying levels of granularity, depending on the task. The table contains statistics on the number of forms and senses for each of the parts-of-speech represented in WordNet, and also sense/subsense information.

Table 3. *Sense statistics for words in WordNet and knoWitiary*

POS	**WordNet (EN)**			**knoWitiary (EN)**				
	# forms		# senses	# forms		# senses	# subsenses	
adj	21,499	30,070	(1.398)	91,218	110,179	(1.208)	110,478	(1.211)
adv	4,475 5,	592	(1.25)	15,251	17,397	(1.141)	17,435	(1.143)
noun	117,953	146,512	(1.242)	378,206	457,147	(1.208)	459,870	(1.215)
verb	11,540	25,061	(2.171)	92,589	116,914	(1.263)	117,759	(1.272)
total	155,467	207,235	(1.33)	577,264	701637	(1.215)	705,542	(1.222)
POS	**WordNet (IT)**			**knoWitiary (IT)**				
	# forms		# senses	# forms		# senses	# subsenses	
adj	5,074	6,452	(1.271)	63,215	67,399	(1.066)	67,415	(1.066)
adv	1,634	2,250	(1.376)	3,996	4,915	(1.23)	4,915	(1.23)
noun	34,935	49,219	(1.408)	95,059	108,743	(1.144)	108,762	(1.144)
verb	4,969	9,875	(1.987)	366,138	374,301	(1.022)	374,303	(1.022)
total	46,612	67796	(1.454)	528,408	555,358	(1.051)	555,395	(1.051)

Table 4 includes statistics on the frequency of the different parts-of-speech in Wiktionary, and the overlap with WordNet (the English 3.1 and Italian versions) for the POS classes represented in WordNet: adjectives, adverbs, nouns and verbs. The high number of word forms for verbs in Italian is caused by the inclusion of inflections. These numerous inflected entries provide a de-facto inflectional derivational resource for Italian. The statistics shown here were obtained from one English Wiktionary dump[10]. Meyer et. al [1] show an overview of Wiktionary coverage (2012), and for three languages (English, German, Russian) extensive comparison between the lexicon from the three Wiktionary versions and the corresponding wordnets, and the coverage of several word lists representing the basic vocabulary of each of the three languages. Meyer et al. [1]'s analysis shows that Wiktionary's coverage of the basic vocabulary is very high, thus supporting its use as a lexical reference resource.

Wiktionary's coverage of WordNet's vocabulary is high, but the majority of Wiktionary entries are not included in WordNet. Even for shared vocabulary, Meyer et al. [3] show that even with a high accuracy sense mapping (estimated on a set of 2,423 pairs of WordNet-Wiktionary senses), more than 370,000 Wiktionary

[10] Version from 10.04.2014: enwiktionary-20141004-pages-articles.xml

senses remain unmapped (on the Wiktionary version used). This shows that when including only the mapped vocabulary and senses, the majority of the potential information to be added is in fact discarded.

Table 4. *Lexical overlap with WordNet*

POS	Overlap	coverage rel. to WN	freq. In EN WordNet	freq. In knoWitiary (EN)
adjective	15,924	74.07%	21,499	91,218
adverb	3,837	85.74%	4,475	15,252
article			–	15
cardinal numeral			–	90
conjunction			–	226
determiner			–	122
interjection			–	2,055
noun	51,283	43.48%	117,953	378,212
numeral			–	200
participle			–	3
preposition			–	501
pronoun			–	441
suffix			–	644
verb	9,837	85.24%	11,540	92,590
total	80,881	52.02%	155,467	581,586

Lexical overlap with Italian WordNet

POS	Overlap	coverage rel. to WN	freq. In IT WordNet	freq. In knoWitiary (IT)
adjective	4,119	81.18%	5,074	63,215
adverb	1,386	84.82%	1,634	3,996
article			–	12
conjunction			–	158
interjection			–	202
noun	21,273	60.89%	34,935	95,050
numeral			–	1
participle			–	4
preposition			–	312
pronoun			–	210
suffix			–	322
verb	4,260	85.73%	4,969	366,139
total	31,038	66.59%	46,612	529,630

4.2. *Relation comparison*

Relations such as SYNONYMS, HYPERNYMS, HYPONYMS, ANTONYMS appear in both WordNet and Wiktionary (in

Wiktionary they appear as sections, which we then formalized as relations), but other relations are particular to one or the other of the resources. The way they are encoded is also different. In Wiktionary, most of the relations presented here (except the etymological ones) appear as section headers, with the related terms presented as a list. In WordNet most relations are between synsets, with the relation of a word belonging to a certain synset explicit through index files. The synonymy relation is implicit between words belonging to the same synset. Relations between synsets can apply to all or specific elements of the synset, and this is signaled within the data files. When computing the number of instances for each WordNet relation this will be taken into account to obtain the correct number of relations in the resource. For the comparison all relation extracted from Wiktionary are named, and from WordNet the 10 most frequent relations are considered separately[11], with all the others grouped under OTHER.

The low overlap in terms of relations show that Wiktionary covers qualitatively different information than WordNet. In the previous work involving Wiktionary and mapping it onto WordNet, the focus is on the mapping of word senses. Had the relation between the mapped word senses been also added, the enrichment in this respect would have been small compared to the original size of the resource: 77,548 pairs of English words (not senses, though) appear in Wiktionary and are connected through a relation, and appear but are not connected in WordNet. Compared with WordNet 3.1's original size (1 million+ relations), the increase is small. The real richness would have come from additional entries that could not be mapped, and their relations.

[11] In WordNet there are three types of meronym and holonym relations. While it is a useful distinction, for the statistics we use only the coarser MERONYM/HOLONYM.

Table 5. *Relation overlap between knoWitiary and English WordNet (3.1)*

Relation overlap: English WordNet → knoWitiary mapping					
knoWit. relation	overlap	overlap with homonym. relation	highest overlap (*WN* rel.)	freq. in WordNet	freq. in knoWitiary
ABBRV	–	–	–	–	179
ACRONYM OF	–	–	–	–	502
ALTERNATIVE FORMS	3,009	–	SYNONYMS (2,882)	–	53,229
ANAGRAMS	239	–	SYNONYMS (192)	–	114,743
ANTONYMS	2,296	2,167	ANTONYMS (2,167)	7,983	18,792
BORROWING	–	–	–	–	805
COGNATE	–	–	–	–	4
COMPOUND	–	–	–	–	11,341
COMPOUNDS	–	–	–	–	8
CONJUGATION OF	–	–	–	–	3
CONFIX	–	–	–	–	2,964
DERIVED TERMS	9,728	3,968	DERIVED TERMS (3,968)	74,680	126,317
DESCENDANTS	8	–	DERIVED TERMS (5)	–	273
ETYM	–	–	–	–	47,684
ETYMTWIN	–	–	–	–	6
HOLONYMS	67	45	HOLONYMS (45)	103,246	377
HYPERNYMS	766	597	HYPERNYMS (597)	364,600	6,661
HYPONYMS	766	597	HYPONYMS (597)	364,600	6,664
MERONYMS	67	45	MERONYMS (45)	103,246	377
RELATED	13,336	–	DERIVED TERMS (7,577)	137,209	111,995
SEE ALSO	3,781	23	SYNONYMS (1,161)	4,732	57,827
SYNONYMS	27,609	14,570	SYNONYMS (14,570)	157,394	149,645
total	61,672	22,012		1,180,481	710,396
Relation overlap: knoWitiary → English WordNet mapping					
WN relation	overlap	overlap with homonym. relation	highest overlap (*knoWit* rel.)	freq. in WordNet	freq. in knoWitiary
ANTONYMS	2,683	2,167	ANTONYMS (2,167)	7,983	18,792
ATTRIBUTE	406	–	RELATED (236)	3,418	–
DERIVED FROM/PERTAINYM	1,230	–	RELATED (1,108)	8,074	–
DERIVED TERMS	12,177	3,968	DERIVED TERMS (3,968)	74,680	126,317
HOLONYMS	881	45	SYNONYMS (346)	103,246	337
HYPERNYMS	6,184	597	SYNONYMS (4,081)	364,600	6,661
HYPONYMS	10,580	597	SYNONYMS (4,081)	364,600	6,664
MERONYMS	953	45	SYNONYMS (346)	103,246	377
OTHER	4,097	–	SYNONYMS (3,284)	137,209	111,995
SEE ALSO	546	23	SYNONYMS (473)	4,732	57,827
SYNONYMS	21,952	14,570	SYNONYMS (14,570)	157,394	149,645
total	61,672	22,012		1,191,973	478,655

Table 6. *Overlap with the Italian WordNet for the Italian entries in knoWitiary*

Relation overlap: English WordNet → knoWitiary mapping					
knoWit. relation	overlap	overlap with homonym. relation	highest overlap	freq. in WordNet	freq. in knoWitiary
ABBRV	–	–	–	–	5
ACRONYM OF	–	–	–	–	1
ALTERNATIVE FORMS	205	–	SYNONYMS (172)	–	1,598
ANAGRAMS	16	–	SYNONYMS (10)	–	194,225
ANTONYMS	24	2	ATTR./HYPON./ HYPERN (6)	22	3,017
BORROWING	–	–	–	–	114
COMPOUND	–	–	–	–	284
CONJUGATION OF	–	–	–	–	294,688
CONFIX	–	–	–	–	2,998
DERIVED TERMS	329	–	HYPONYMS (182)	–	14,045
DESCENDANTS	2	–	HYPON./HYPERN (1)	–	125
ETYM	–	–	–	–	7,664
HYPERNYMS	9	8	HYPERNYMS (8)	116,318	22
HYPONYMS	9	8	HYPONYMS (8)	116,318	22
RELATED	2,060	–	SYNONYMS (714)	6,909	118,669
SEE ALSO	288	–	SYNONYMS (113)	–	3,410
SYNONYMS	5,852	3,650	SYNONYMS (3,650)	53,153	30,802
total	8,794	3,668		292,720	671,689
Relation overlap: knoWitiary → English WordNet mapping					
WN relation	overlap	overlap with homonym. relation	highest overlap	freq. in WordNet	freq. in knoWitiary
ANTONYMS	2	2	–	22	3,017
ATTRIBUTE	149	–	RELATED (126)	3,372	–
HOLONYMS	143	–	RELATED (96)	8,429	–
HYPERNYMS	1,593	8	SYNONYMS (998)	116,318	22
HYPONYMS	1,767	8	SYNONYMS (998)	116,318	22
MERONYMS	144	–	RELATED (96)	8,429	–
OTHER	263	–	SYNONYMS (170)	6,909	118,669
SYNONYMS	4,733	3,650	SYNONYMS (3,650)	53,153	30,802
total	8,794	3,668		311,851	152,532

5. NOVEL PERSPECTIVES ON NLP TASKS

knoWitiary contains much lexical information that is novel, and combines pieces of information that have previously not been accessible from a single resource.

Etymology, in particular, has proved its usefulness in bridging different languages in a cross-lingual text categorization task [17]. Cross-lingual text categorization consists in categorizing documents in a target language L_t using a model built through supervised learning on a labeled dataset in source language L_s. The task is even more difficult when the datasets in the two languages are not parallel (there is a 1:1 mapping

between the texts contained in the two datasets, one being the translation of the other), but rather consist of comparable corpora (i.e. documents on the same topics, such as sports, economy). Etymological relations provide a layer of shared word-ancestors that connect the two languages, thus allowing the model to capture text-category associations at this shared linguistic level. Having translations also available would allow this bridge to be further enriched, thus leading to better shared models across the languages.

Etymological information is also crucial in studying the evolution of language during different epochs. It could be possible to study why some words evolve rapidly through time while others stay the same, often with an identical meaning in many different languages. We could verify hypotheses such as: the more often a word is used, the less likely it is to mutate. A similar observation was made relative to verbs, where those that are most commonly used have irregular forms [21].

Instances of compounding and word derivation, in particular in those situations where the resulting term is not compositional (or not any longer) – e.g. $breakfast$ are instances of language creativity that can be further studied to find how such term have been generated, and thus endow a machine with similar capabilities. Measuring the semantic distance between a term and its etymological children could show how metaphors are coined, and what kind of word relations have been used to make this creative jump. The connected nature of Wiktionary would allow for such investigations.

Other creative language tasks would benefit from a rich lexical resource. For example [22] propose a computational approach to generate neologisms consisting of homophonic puns and metaphors based on the category of the service to be named and the properties to be underlined. This kind of task is very challenging from a lexical knowledge point of view, because it requires a combination of semantic, phonetic, lexical and morphological knowledge to automatize the process.

6. CONCLUSIONS

The work presented here had two motivations: (i) to obtain a coherent and consistent lexical resource that contains as much information as possible about words and their relations, (ii) to measure what could be gain and what would be lost by forcing a mapping of such a resource onto another structure. To obtain the lexical resource we processed Wiktionary, the on-line collaboratively built dictionary covering a treasure trove of entries and relations in numerous languages. To measure this against other resources used in NLP, we choose WordNet, since it is the most frequently used, and also the base for mapping other resources, including those based on Wiktionary itself. We have explored both the lexical and relation overlap, which shows that Wiktionary provides a different kind of information than WordNet does. Mapping it onto WordNet would mean discarding such unique information, with unknown impact on both the tasks that use the mapped version, and on tasks that are never attempted because the mapped resource lacks the needed information/links.

REFERENCES

1. Meyer, C. M. & Gurevych, I. 2012. Wiktionary: A new rival for expert-built lexicons? exploring the possibilities of collaborative lexicography. In S. Granger & M. Paquot (Eds.), *Electronic Lexicography* (pp. 259-291). Oxford: Oxford University Press.
2. Miller, T. & Gurevych, I. 2014. WordNet-Wikipedia-Wiktionary: Construction of a threeway alignment. In proceedings of the *9th Language Resources and Evaluation Conference (LREC 2014)*.
3. Meyer, C. M. & Gurevych, I. 2011. What psycholinguists know about Chemistry: Aligning Wiktionary and WordNet for increased domain coverage. In proceedings of the *5th International Joint Conference on Natural Language Processing* (pp. 883-892).
4. Fellbaum, C. (ed.) 1998. *WordNet: An Electronic Lexical Database*. Cambridge, Mass: MIT Press.
5. Mihalcea, R. & Moldovan, D. 2001. Automatic generation of a coarse grained WordNet. In proceedings of *NAACL Workshop on WordNet and Other Lexical Resources*, Pittsburgh, PA, Jun.

6. Vossen, P. (ed.) 1998. *EuroWordNet: A Multilingual Database with Lexical Semantic Networks*. Kluwer, Dordrecht, The Netherlands.
7. Strapparava, C., Gliozzo, A. & Giuliano, C. 2004. Pattern abstraction and term similarity for word sense disambiguation. In proceedings of the *3rd International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (SENSEVAL-3) at ACL-04* (pp. 229-234), Barcelona, Spain, 25-26 Jul.
8. Baccianella, S., Esuli, A. & Sebastiani, F. 2010. SentWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In proceedings of the *7th International Conference on Language Resources and Evaluation* (pp. 2200-2204), La Valetta, Malta, 17-23 May.
9. Magnini, B., Strapparava, C., Pezzulo, G. & Gliozzo, A. 2002. The role of domain information in word sense disambiguation. *Natural Language Engineering*, 8, 359-373.
10. Snow, R., Jurafsky, D. & Ng, A. Y. 2006. Semantic taxonomy induction from heterogeneous evidence. In proceedings of the *21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics* (pp. 801-808), Sydney, Australia, 17-21 Jul.
11. Boyd-Graber, J., Fellbaum, C., Osherson, D. & Schapire, R. 2006. Adding dense, weighted connections to WordNet. In proceedings of the *Third Global WordNet Meeting*, Jeju Island, Korea, Jan.
12. Ruiz-Casado, M., Alfonseca, E. & Castells, P. 2005. Automatic extraction of semantic relationships for WordNet by means of pattern learning from Wikipedia. In proc. of the *10th International Conference on Applications of Natural Language to Information Systems* (pp. 67-79).
13. Zesch, T., M uller, C. & Gurevych, I. 2008. Using Wiktionary for computing semantic relatedness. In proceedings of the *23rd Conference on the Advancement of Artificial Intelligence* (pp. 861-867), Chicago, Ill., 13-17 Jul.
14. Etzioni, O., Reiter, K., Sonderland, S. & Sammer, M. 2007. Lexical translation with application to image search on the web. In proceedings of the *Machine Translation Summit XI*, Copenhagen, Denmark.
15. Richman, A. E. & Schone, P. 2008. Mining wiki resources for multilingual named entity recognition. In proceedings of the *46th Annual Meeting of the Association for Computational Linguistics:*

Human Language Technologies (pp. 1-9), Columbus, Ohio, 15-20 Jun.
16. Navarro, E., Sajous, F., Gaume, B., Prevot, L., ShuKai, H., Tzu-Yi, K., Magistry, P. & Chu-Ren, H. 2009. Wiktionary and nlp: Improving synonymy networks. In proceedings of the *2009 Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources* (pp. 19-27).
17. Nastase, V. & Strapparava, C. 2013. Bridging languages through etymology: The case of cross language text categorization. In proceedings of the *51st Annual Meeting of the Association for Computational Linguistics* (pp. 651-659), Sofia, Bulgaria.
18. de Melo, G. 2014. Etymological Wordnet: Tracing the history of words. In proceedings of the *9th Language Resources and Evaluation Conference (LREC 2014)*.
19. de Melo, G. & Weikum, G. 2010 .Towards universal multilingual knowledge bases. In *Principles, Construction, and Applications of Multilingual Wordnets*. *Proceedings of the 5th Global WordNet Conference (GWC 2010)* (pp. 149-156), New Delhi, India.
20. Gurevych, I., Eckle-Kohler, J., Hartmann, S., Matuschek, M., Meyer, C. M. & Nghiem, T. D. 2013. Uby – a large-scale lexical-semantic resource. In *Book of Abstracts of the 23rd Meeting of Computational Linguistics in the Netherlands: CLIN 2013* (p. 81)
21. Pinker, S. 1999. *Words and Rules*. Canada: Harper Collins.
22. Özbal, G. & Strapparava, C. 2012. A computational approach to automatize creative naming. In proceedings of the *50th annual meeting of the Association of Comptational Linguistics (ACL-2012)* (pp. 703-711), Jeju Island, Korea.

VIVI NASTASE
FBK-IRST, TRENTO, ITALY.
E-MAIL: <NASTASE@FBK.EU>

CARLO STRAPPARAVA
FBK-IRST, TRENTO, ITALY.
E-MAI: <STRAPPAG@FBK.EU>

International Journal of Computational Linguistics and Applications, Vol. 6, No. 2, 2015, pp. 83–103
Received 24/01/2015, Accepted 27/02/2015, Final 20/12/2015.
ISSN 0976-0962, http://ijcla.bahripublications.com

A Romanian Dependency Treebank

CĂTĂLINA MĂRĂNDUC
Alexandru Ioan Cuza University of Iasi, Romania
Academic Institute of Linguistics, Bucharest, Romania

CENEL AUGUSTO PEREZ
Alexandru Ioan Cuza University of Iasi, Romania

ABSTRACT

The Romanian Treebank was created with manual and automatic manually checked annotation. The syntactic relationships were meticulously defined. We aim to affiliate our Treebank to Universal Dependencies, in this way some categories would become subclassifications. For the creation of this Treebank, we have built an annotation interface and a Romanian language dependent parser that works with statistical methods and whose accuracy is not satisfactory. This is why we intend to create another hybrid and rule-based parser. Its programming would include syntactic and semantic background for most verbs in the Romanian language. They will be extracted from Treebank and RoWN (lined up with PWN). For the missing verbs, we will introduce in the Treebank a big number of quotations from eDTLR (Romanian Thesaurus Dictionary). We will add to the Treebank a new layer with semantic annotation based on accurate criteria, starting with RoWN's, to which we will add interdictions. Another derived project is the creation of a multilingual aligned Treebank.

Keywords: Relationship, rules-based parser, semantic background, syntactic background, tree alignment, universal dependencies.

1. INTRODUCTION

Recent comparative studies have shown that besides English, all the other European languages have an insufficient degree of computerization. In terms of vocabulary, Romanian has an average level of computerization, by virtue of the existence of RoWN (Romanian WordNet) [3] aligned with PWN (Princeton WordNet), but it is ranked among the last when regarding the existence of annotated corpora approachable by researchers.

We intended to create a new Treebank for Romanian that would meet the urgent need of the computerization of Romanian language. The creation of the Treebank has started within a joint project shared by the Institute of Information Managementof the Romanian Academy and the Computer Science Faculty ofAl. I. Cuza University of Iasi. It started with a collection of 600 sentences annotated at the syntactical level. During the research in his PhD thesis, Augusto Perez [11] had extended it to 4,600 sentences in December 2014. This Treebank, called UAIC-RoDepTb, is balanced, containing complex sentences from all language registers. Some of Sentences are complex, with a varying extent, starting from 4 components to over 100.The total number of words and punctuation elements reached over 105000[1].

In these structures, using annotation conventions of the dependency grammar type, suited to the characteristics of the Romanian language, there were annotated a great number of relationships, whose names are similar to those of classical grammar. In addition to the verbal mandatory arguments, there were annotated a large number of optional circumstantial relations (called adjuncts or modifiers and generally no classified in the syntactic ontologies), which are useful for further semantic, pragmatic, discursive type of research.

[1] The data in this section were available for April 2015, when the article was communicated to CICLing conference. A recent assessment shows that the UAIC-RoDepTb reached 10,920 Sentences, and 200,764 words and punctuation elements.

2. CORPUS DESCRIPTION

2.1. *Dimensions of corpus*

Unlike the other Treebankfor Romanian, made at the Faculty of Mathematics, University of Bucharest [8], and which had a comparably similar number of trees, our corpus has three times more syntactic units, which demonstrates that we have obtained a superior Treebank regarding the length and complexity of the syntactic structures.

Annotation was performed using an interface called the TreeAnnotator,[2] which allows viewing the arcs between node words or punctuation signs of the tree and inscribing the logos of syntactic relations as arcs labels. The graph obtained has each node positioned above the word from the subjacent initial array of signs;the morphological analysis label is visible as a result of automatic annotation of one of the two POS-taggers for Romanian language.[3]

---

[2] The interface is developed in the master degree paper by IustinDornescu.

[3] TTL POS-tagger from http://www.racai.ro/tools/text and UAIC POS-tagger from <http://nlptools.infoiasi.ro/Resources.jsp> are used from the morphologic previous annotation of sentences in the UAIC-RoDepTb.

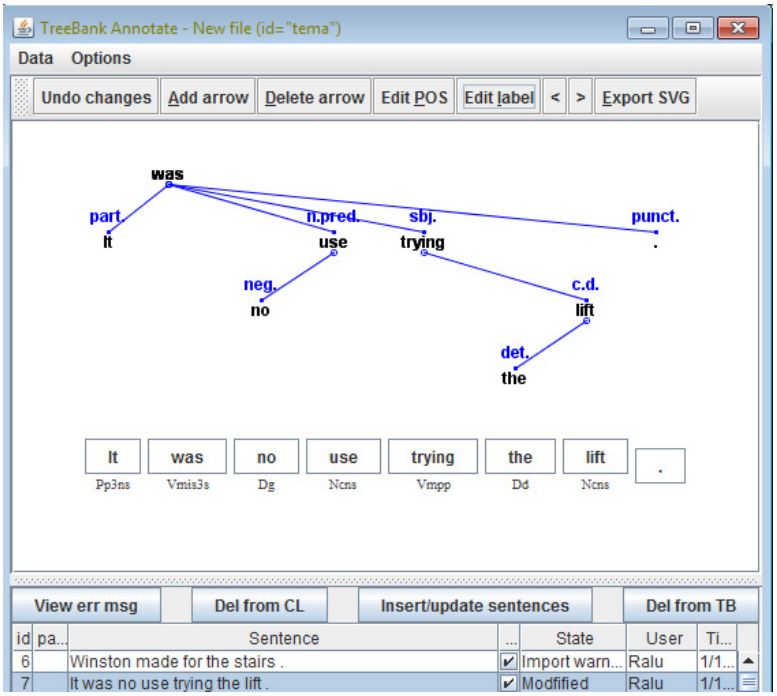


Figure 1. *A graphviewedby the TreeAnnotator framework*

The annotation conventions used were determined by comparing the two language expert annotators' options and choosing the mutually agreed solutions in accordance with the language distinctiveness and the complexity of natural language phenomena that we were confronted with.

2.2. *Annotation conventions*

Hence an establishment was reached regarding a coherent system of punctuation annotation which, except commas marking the coordination, is subordinated to the head of the sub-tree that is isolated from the rest of the sentence by inverted commas, parentheses or commas. It may be an exogene structure, which comes from another emitter citation, which is grouped around a vocative without syntactic function, an optional dependency of a verb or an apposition. It can be seen that the punctuation

annotation, imposed by the conventions of dependency Grammar theories, is justified as the punctuation signs have a defined syntactic-semantic role.

Another specific annotation convention regards the subordinate elements of elliptical regents. Our solution is different from that envisioned on the Universal Dependencies[13] site,and we consider it better. Example:

John won bronze, Mary silver, and Sandy gold. (1)

Their solution is to mark a relation (labeled *remnant*) in fact non-existent, between the three subjects on the one hand, and the three complements on the other hand (see Figure 2). In fact, they are part of different sentences and relations are established only between each subject and its object.

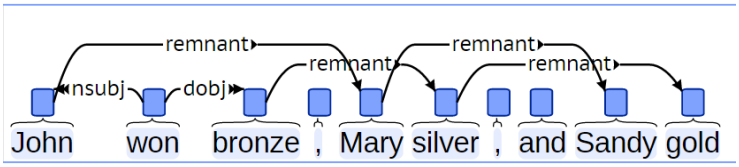


Figure 2. *Annotation of eliptics regents in English Treebank affiliated of Universal Dependencies*

We chose to use the label *remnant* for the relationship between the verb and the coordination elements, which in this sentence replace the elliptic regents, borrowing the properties of the root and becoming heads of pairs two and three of *subj* and *dobj* related (see Figure 3).

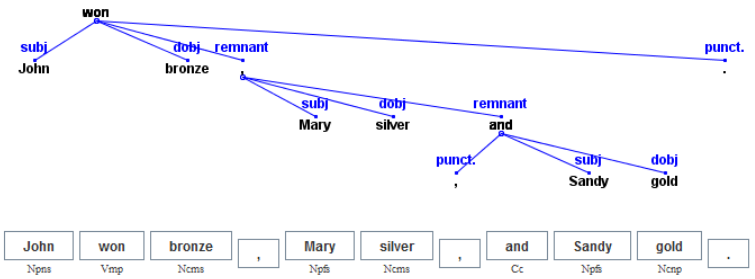


Figure 3. *Annotation of eliptics regents using Romanian Treebank conventions*

2.3. *Parsers*

For the automatic annotation, we used a FDG type parser, which was trained using the first manually annotated 1000 sentences, then with the bootstrapping method the automatically annotated sentences was manually corrected and the training was resumed with a gold corpus increased to 2000 and 3000 phrases. We then used another parser, similarly trained.[4] The evaluation ofparsers led to modest results (see Table 1 and 2). Both parsers developed variants of parser for Romanian built by J. Nivre [6-7].

In their article, Călăcean and Nivre [2] described asyntactic annotated corpus which resulted from the operation of their parser, which has 4,042 sentences with 36,150 tokens, punctuation excluded. The labeled attachment score for this parser had 88.6% accuracy, and the unlabeled attachment score had 92.0% accuracy for this corpus. The corpus includes short sentences with a simple structure, in a unique style, with political and administrative journalistic topics. Texts including complex ambiguities were avoided as much as possible and removed from their Treebank.

The evaluation of the two parsers on our Treebank led to the following modest results, although heuristic modules were used to improve the parser built by J. Nivre:

[4] FDG parser was developed in the graduation degree paper by Claudius Popa. The other parser was built by RaduSimionescu and can be found at http://nlptools.infoiasi.ro/WebFdgRo/.

Table 1. *FDG parser evaluation*

Metrics	Good trees	Label precision	Head precision	Both precision
Score	3,20%	55,95%	65,32%	61,03%

Table 2. *Evaluation of nlptools.uaic parser*

Metrics	Label precision	Head precision	Both precision
Score	62,02%	68,88%	58,56%

The poor results can be explained by the small size of the training corpus, the stylistic variety and the complexity of syntactic constructions. While other corpora have simple and typical examples specifically selected for an efficient training of the parser, the authors of the present Treebank, being both linguists, are concerned with the formulation of annotation conventions thorough and flexible enough to illustrate a wide variety of natural language and specific for Romanian linguistic facts.

The corpus illustrates the legal style, including texts of Acquiscommunitaire, Romanian and universal fictional texts, and also the journalistic style of Frame-Net translated into Romanian. The result of manual annotation can sometimes be the difference between relationships interpreted by the two annotators. We had not rigorously assessed the agreement between annotators using a statistical method. We preferred proceeding from mutual corrections and debate to establish a common view.

To get bigger corpus drive, would require a more effective automatic annotation, and to increase the accuracy of parser would require a bigger gold corpus for training. The two shortcomings (the reduced size of the corpus and the modest parser accuracy) are correlated. UAIC-RoDepTb has been developed too much through manual correcting of automatic parsing. In addition, there is an increasing interest among researchers, especially linguists, in the creation of processing tools for old Romanian, i.e., sixteenth and seventeenth centuries, Romanian' and this task would cause further difficulties for the training of a parser based on statistical methods. In the next stage of the research, we propose building a hybrid, statistic and rule-

based parser, trained in distinct modules for contemporary and for old Romanian.

3. THE UNIVERSAL DEPENDENCY TREEBANKSFORMAT

3.1. *The need to unify the format of resources*

To make comparisons, to permit the reuse of our resources in various kinds of research, to get good results in our research, we aim at affiliating our Treebank to the Universal Dependencies group, founded in 2013, and to which were affiliated dependency Treebank corpora for 30 languages [13][5]. The categories of annotated relationships will be automatically translated into the categories proposed by this project, so some of our annotations will remain in a different layer of annotation. We will continue the process of annotation using the new labels for categories and subcategories.

To be implemented from a system of annotation to another, categories of the two systems should be in a relationship of equivalence or inclusion. There are cases in which the categories of our annotation system are in a relationship of intersection with Universal Dependencies ones.

We have carefully studied the annotation conventions in the English Treebank, which relationships were taken as examples on this site so as to compare them with those used by us. Each type has an intension, a definition of the relationship, and an extension, represented by the set of natural language facts that can be annotated with that relationship. It is important to study the relationship established between the set of relations defined by UD (Universal Stanford Dependencies) and the relations circumscribed by the UAIC-RoDep Tb conventions.

At a first glance, it would seem that the transposition will be easy, because there are many cases of equivalence between logos, or double inclusion between their extensions, for example: {appos}≡{ap.} {neg}≡{neg.} {parataxis}≡{incid.} {mark}≡{part.} {punct}≡{punct.} {vocative}≡{voc.}. But these are not the most important sections of the system. In other cases,

[5] http://universaldependencies.github.io/docs/#language.

there are differences between the sets of phenomena which are annotated. In other cases the sets are in inclusion or intersection relation, which is explained by the existence of theoretical and systemic differences that we try to synthesize and to comment.

3.2. *The distinct annotation of subordinate clauses*

The first observation is that, although clearly intended to establish a minimum number of syntactic categories, the same relationships are annotated differently when establishing a lexeme with his regent to where they are established between the same propositional and construction regent.

Examples:

*Cuvintele*lui au sens./His *words* make sense. (2)
Ceeacespune el are sens./ *What he said* make sense. (3)

In sentences (1, 2) we annotate the same relationship, sbj. on the line joining the underlined sequences of their regent. In contrast, according to the conventions of UD annotation, in sentence (2) the underlined word make *nsubj* relationship with the verb and in sentence (3), the relation of the underlined sequence is annotated with the *csubj* relationship (clause-subject). In our opinion, this is the same relationship with the same meaning and can be easily replaced with each other in any context. We notice that the label sbj. of our Treebank has an extension covers a lot of linguistic phenomena, and include the two sets referring to labels *csubj* and *nsubj* of UD annotation system (4). The same types of relationship are between the sets of expressions (5), due to the fact that our logo aux. is used in annotating to all types of auxiliaries.

{nsubj}⊂{sbj.}; {csubj}⊂{sbj.} (4)
{aux}⊂{aux.}; {auxpass}⊂{aux.} (5)

3.3. *Types of circumstantial modifiers*

As already mentioned, in our Treebank, which aims to further the basis for semantic annotation and discursive, argumentative and

pragmatic research, great attention was given to establishing the types of Circumstantial Modifiers. They are annotated in UD system indiscriminately, without regard for their particular purposes and argumentative report that is set by the regent, placing emphasis instead on their morphological peculiarities, which however results from the previous automatic annotation with the POS-tagger. We do not know what theoretical arguments justify a return to the inferior morphologic level, since higher syntactic level should look, we believe, at more complex levels of the communicative organization including semantic, textual and discursive information.

Our system includes 14 categories: c.c.m. (modal circumstantial), c.c.t. (temporal circumstantial), c.c.l. (local circumstantial), c.c.cond. (conditional circumstantial), c.c.scop. (purpose circumstantial), c.c.cz.(cause circumstantial), c.c.cons. (consecutive or result circumstantial), c.c.conc. (concessive circumstantial), c.c.exc. (exception circumstantial), c.c.instr. (instrumental circumstantial), c.c.soc. (associative circumstantial), c.c.cumul. (cumulative circumstantial), c.c.opoz. (opposition circumstantial), c.c.rel. (relational circumstantial), categories of relationships that are interesting for further research and we do not intend to give up these distinctions once established and annotated; they will be kept in a different layer of annotation. To establish convergence with UD annotation system, as will be seen in what follows, circumstantial relations expressed by adverbs (i.e., adjuncts) are classified as *advmod*, while those expressed by nouns with a preposition to be annotated as *pmod* and those expressed by sentences are annotated with *clmod*. Consequently, a complicated system will result, with 14 intersections of each of the three types below: 14⋆3=52intersectionsbetweencategories(6-8):

{advmod}∩{c.c.m.} ... {advmod}∩{c.c.rel.}; (6)
{advcl}∩{c.c.m.} ... {advcl}∩{c.c.rel.}; (7)
{pmod}∩{c.c.m.} ... {pmod}∩{c.c.rel.}. (8)

The system proposed by UD relations will use mandatory dependencies of really important verbs (called arguments),

annotated with *nsubj*, *dobj*, *iobj*, which would be added *secobj* (secondary) *andagc* (Agent argument). Verbal circumstantial dependencies (called adjuncts) are considered optional, therefore less important, not reaching the basic structure of the sentence. But there are verbs for which certain circumstantial "adjuncts" are mandatory, which is another argument in favor of the syntactic-semantic significance of these categories. Examples:

A se deplasa de la Ana la Caiafa./
To move from Ana to Caiaphas. (9)

A dura de la oraunupână la oracinci./
To last for one hour at five. (10)

Camionulcântăreştepatru tone./
The lorry weighs four tons. (11)

In the Example (9), the verb *a se deplasa* "to move" involves two limits of space, so it has two mandatory dependencies c.c.l. In the Example (10), the verb a dura "totake" involves two time limits, so it has two mandatory dependencies c.c.t. In the Example (11), the verb a *cântări* "to weigh" has c.c.m. quantitative mandatory.

3.4. *Noun modifiers*

A similar situation emerges if we study the modifiers of the noun annotation in the two systems. The annotation conventions of UD can be: *amod*, *acl*, *nmod*, *pmod*, noting that the last category ascribes that dependence on verb. The focus is on annotation relationships based on their morphological realization, although, in our opinion, it is syntactically less important and in addition, marking this information is redundant, ranging in annotation POS tagger, preceding syntax.

In our annotation system, the noun dependents are not differentiated according to whether or not there exists a preposition, leading to junctions (13). The noun dependents are: a.adj. (adjectival attribute), a.subst. (noun attribute), a.pron. (pronominal attribute), a.adv. (adverbial attribute), a.vb. (verbal attribute). To make the transposition of noun modifiers from one

format to another, again you have to take into consideration a number of relations of inclusion and intersection. In the system of UAIC-RoDepTb, *a.adj*. modifier includes numerals and determiners derived from pronouns, whereas *det*. (categories of determination) content only articles, whwn in the UD system *det* includes determiners derived from pronouns (14):

$$\{\text{nummod}\} \subset \{\text{a.adj.}\} \quad \{\text{det}\} \cap \{\text{a.adj.}\} \quad \{\text{amod}\} \subset \{\text{a.adj.}\} \quad \{\text{a.adv.}\} \subset \{\text{advmod}\}; \tag{12}$$

$$\{\text{a.pron.}\} \cap \{\text{nmod}\} \quad \{\text{a.subst.}\} \cap \{\text{nmod}\} \quad \{\text{a.pron.}\} \cap \{\text{pmod}\} \quad \{\text{a.subst.}\} \cap \{\text{pmod}\}. \tag{13}$$

These situations can be resolved automatically just because the information is already morphologically annotated and it is likely that changes can be made without loss of information and without the need for another different layer of annotation, as in the case of complements.

3.5. *Annotation of prepositions and conjunctions*

UD uses annotation conventionsin which the conjunctions, prepositions, and marks of coordination are not considered head for the words that these connectors entered in the text. In our system, copulative verbs, prepositions and conjunctions are considered head, but this convention may be changed without great loss of information unless, except the case of examples similar with (1), in which the conjunctions or punctuation elements are instead of elliptical regents (translating by coordination their information).

But what seems highly inappropriate is the fact that the preposition annotation system is labeled UD case, because in some languages, (as French) a preposition forms the genitive case. It would be better to annotate it as mark. Prepositions introduce highly specialized semantic relations which are expressed not only by nouns or pronouns, but also by other parts of speech, like verbs, adverbs, for which the case category is not appropriate. In Romanian, the preposition does not expresses the case of nouns, but it requires, as a true regent, a specific noun

case, and inflected forms are expressed by the enclitic definite article. There are illustrative examples:

El se ascundedupăcoteț./ He is hiding behind the cage. (14)
El se ascundeînapoiacotețului./ He is hiding behind the cage. (15)

Examples (14, 15) are synonyms. In (14), the preposition *după* "after" requires the accusative case and inarticulate form of noun. In (16), another preposition, *înapoia* "behind", requires the genitive case and articulate form of noun. The case is formed by the definite article *-ului* and not by the preposition. The label of this relationship should be, in our opinion, prep, or, if we decide to unify prepositions with subordinate conjunctions, *subord*.

Universal Conventions annotation should believe, to be the result of laborious consultations between computer scientists and linguists specialized in different natural languages, and not to unify categories of relations sacrificing semantic information in favor of the morphological one. It does not contain generalizations of some specific English features for all languages, such as those related to the expression of genitive case in French or the lack of reflexive in English. But since the unification of terminology and annotation formats is a high importance, we will submit the majority consensus and we adopt the system of relations of UD.

4. USING OUR TREEBANK FOR BUILDING NEW RESOURCES

4.1. *Inventory of predicate argument structures*

As we mentioned, we design the Treebank as a basis for more complex annotations. Outside to the standardization of the format and the increase of Treebank's size, we propose to create, through reuse of this corpus, some new resources for Romanian.

One of the development possibilities will be Treebank's enrichment through automatic annotation of semantic information extracted from RoWN.[6] This information will be assigned according to the lemma of the word, which was

[6] http://www.racai.ro/wnbrowser/

automatically annotated previously the syntactic annotation. Semantic annotations will be then corrected by experts.

After this preliminary stage, we use our Treebank to the development of a resource consisting of an inventory as comprehensively as possible of predicates arguments and adjuncts structure for Romanian. The predicates with their necessary or facultative dependencies will be extracted from the Treebank already annotated with syntactic and semantic labels. The corpus began to be created by computer scientists from RACAI (Romanian Academy Research Institute for Artificial Intelligence) by extracting such verbal patterns from RoWN. These types of syntactic-semantic structures [1] have been extracted for more than 500 verbs. Here's an example:

{mânca} nom*AG(person:1|animal:1)=acc*SUBSTANCE(food:1) (16)

The subject of the verb *amânca* "to eat" in (18), has the semantic role AG (ent), that can be satisfied by *animal* with meaning 1 in RoWN, by *person* with the meaning 1 in RoWN, or any noun in the nominative case (nom), which appears in RoWN as a hyponym of *person*: 1 or of *animal*: 1. The direct object in the accusative case (acc) has the semantic role SUBSTANCE and can be satisfied by the noun *hrană* "food" with the meaning 1 or by any of its hyponyms in RoWN. In parallel, at UAIC were built RoVerb-net, by the adoption of English Verb-net and by searching in Romanian corresponding examples for its categories of verbs [5]. The semantic roles of verbal group have been annotated by importing the Frame-net system of annotation in [15].

The new corpus, extracted by the Treebank, once it will be sufficiently representative for Romanian language verbs, will be used first in linguistic research, on the other hand in the natural language engineering, for programming a hybrid, statistic and rule-based syntactic parser. It is an absolutely necessary tool for increasing the size of the Treebank.

A long practice of automatic parsing error corrections has led us to the conclusion that the most affected by parsing errors are

the structures containing misinterpretations in the upper place, at the root detection and his arguments required. Free word order in Romanian makes the parser to confuse the subject with the direct object or the predicate name. It is therefore necessary that the structural patterns of predicates contain rules in the form of prohibitions, i.e. syntactic-semantic dependencies that cannot be subordinated to that verb. The reflexive verb cannot have a direct object, for example. The confusion in the relations introduced by prepositions are more numerous in Romanian because there are more features and more occurrences of nouns preceded by the preposition than those with direct connection.

It is important to establish semantic categories as numerous as suitable for the purpose intended and as close to an international standard. In addition to the syntactic categories found in verbal-semantic structures extracted from RoWN, we can use PDEV[7] (Dictionary Pattern of English Verbs) that provides semantic frames for 5,602 verbs. The shape can be seen in Fig. 4. These structures protocols that were used to create the English Verb-net, and also the classification made in [10] can be largely translated into Romanian or can be taken as a model for creating patterns whose structure is not similar in both languages, Romanian and English.

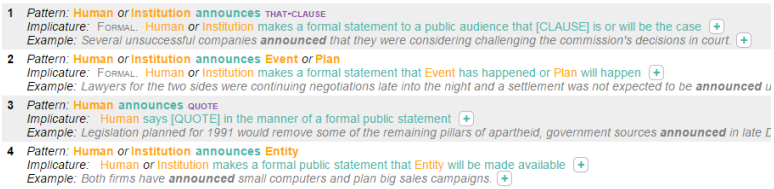


Figure 4. *Syntactic semantic structure of the verb announce in the PDEV*

For the Romanian language, the structure of Figure 4, which add syntactic information will look like this:

[7] http://www.pdev.org.uk/#browse?q=;f=C

{anuţa:1} *nsubj* [HUMAN or INSTITUTION] announces *cs* [CĂ] *dobjcl* [CLAUSE]. Example: (17)
Municipalitateaanunţăcăimpozitelevorcreşte./ The Municipality announce that taxes will increase.
{anuţa:2} *nsubj* [HUMAN or INSTITUTION] announces *dobj* [EVENT or PLAN] *iobj* [HUMAN 2].
Example: Emil anunţăcăsătoriasaprietenilor. / Emile announces his marriage to friends.
{anuţa:3} *nsubj* [HUMAN] announces *dobj* [DATE]
Example: Meteorologulanunţă 32° C. / The weatherman announces 32° C.

The fourth situation in Figure 4 has no direct parallel in Romanian. But as dependency grammar conventions do not provide transformations, we have to include the appropriate distinct patterns as the examples above (17), with the same verb in the passive, reflexive, impersonal voice, if they are present in Romanian. Here's one of them:

{anunţa:4} *nsubjpass* [EVENT or PLAN] is announced *prep* [DE (CĂTRE)'by'] *cag* [HUMAN/INSTITUTION]
Example: Căsătoriaesteanunţată de (către) Emil./ The marriage is announced by Emile. (18)

In a recent article describing a similar resource for Italian [9], the authors show that such syntactic semantic patterns have been extracted by lexicographical methods from large text corpora. So, they detect new possible situations that did not fit the pattern set, such as:

Pattern: [HUMAN1] annuncia [EVENT] a [HUMAN2]
Corpus lines: L'altoparlanteannunciaval'arrivodeltreno/ The speaker announced the arrival of the train. (19)

The authors consider that (19) is a type mismatch, wrong framed as an example to existing types. It would be necessary to introduce a new pattern:

[DISPOSITIF, MACHINE] annuncia [EVENT] a [HUMAN] (20)

Besides the Treebank corpus from which we can extract sentences containing the verb for which we have to build anew pattern structure, we have another resource for Romanian, eDTLR (Electronic Romanian Language Thesaurus Dictionary)[8] obtained in a project in which more Romanian academic institutions participated, by parsing the 32 printed books, scanned, converted into text by OCR-ization and corrected by experts. From another corpus related to the same project, the bibliographical sources of eDTLR, we have already randomly selected 1000 sentences that were introduced in the Treebank and which have the advantage of a balanced selection of the most representative Romanian texts of every style.

The dictionary contains over 2 million quotations arranged chronologically and by the numbers of meanings of every word-entry, following the definition. We have to select from these quotations those related to verbs that do not have syntactic-semantic structures in the RoWN, they will be introduced in the Treebank corpus enriched with semantic annotations. Unlike the examples (17-20), the examples of the new resource will have a tree form, with exact indication of head for mandatory and optional dependencies.

Type prohibition rules attached to patterns of verbs for constructing rules-based syntactic parser are selected from another corpus, unfortunately also rising, i.e. the corpus of automatic syntactic parsing errors corrected by experts. The rules will be obtained by generalization of the most common errors, and they will have the form:

[NOT] *refl* [SE] announces *dobj* [EVENT or DATE]. (21)
Se anunţă o vremefrumoasă. /It announces nice weather.

In the Example (21) in Romanian it is about not a *refl* but an *impers* value and the syntactic function required is the *nsubj* or

[8] Built between 2007 – 2010, in a project financed by Romanian Government and coordinated by UAIC-FII (https://consilr.info.uaic.ro/edtlr/wiki/index.php?title=Digitalizing_the_Thesaurus_Dictionary_of_the_Romanian_Language).

csubj achieved through a sentence, which will be established in other patterns that will result from the corpus of quotations introduced in the Treebank.

4.2. *Aligned corpus 1984.en.ro.fr*

Since our Treebank already contains nearly 1,000 Sentences coming from Romanian version of the project alignment Translations 1984[9] we decided to build a corpus which contained the annotated sentences in the Romanian version of 1984 and the parallel sentences in English and French, annotated with respect of our Treebank conventions. The corpus, began in December 2014 by Master of computational linguistics students of Faculty of Computer Science of Al. I. Cuza University of Iasi, has 250 Sentences in each of the three languages, each containing about 6,500 tokens. Corpus size could be increased by aligning other French and English Sentences from the Romanian already annotated or by adding other languages.

In Figure 1 there is a tree belonging from the aligned English corpus that was annotated too in the Tree Annotator interface used to create the Romanian Dependency Treebank. Fig. 5 shows the tree-structure of sentences aligned with it.

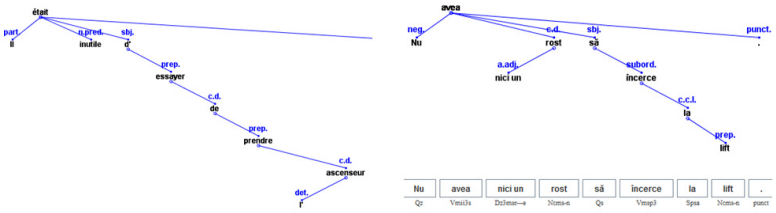


Figure 5. *Trees alined with the tree from Figure. 1.*

This project is important for the EBMTs (Example based machine translations) by introducing tree structures for the language source and for the language target into the memory of

[9] MTE (MultiText-East, morpho-syntactic manually annotated) (nl.ijs.si).

translation software, or to evaluate the quality of automatic translations, or to extract rules for rule-based machine translation. It is required already a program that aligns syntactic relations structures of the trees in different languages and automatically displays the differences found, like the program described in [14].

The study of aligned trees shows us structural syntactic differences between languages and then we can make assumptions on the most appropriate format annotation that would have to use a universal system of categories as the UD. At the same time, we see which format is readily convertible to other format that should have a system of dependency Treebanks. Although permanent concern us the most appropriate formalism to the Romanian languages pecific structures, we should avoid as far as possible the use of convertible difficult relationship in the annotation system conventions of other languages.

5. CONCLUSIONS

In this paper we tried to demonstrate that it is extremely important, especially for a small movement language with scientific research underfunded, but not limited to such a language, to reuse existing resources for building new resources, with a higher degree of complexity of the annotations. For this, we need most often towards compatibility annotation formats new or old. It is necessary to create tools to make unification of annotations, conversions, or simplifications, tools organized in chains of automatic processing. At our faculty such operations are carried out through the hierarchical meta-system of tools and resources ALPE (Automated Linguistic Processing Environment) [4], [12].[10]

In this context, the task of linguists, besides correcting the errors of natural language processing programs, is precisely to carry out studies concerning not only the logos of private correspondence with other standardized universal logos but also about the transposition of resources in a format based on a

[10] http://creativecommons.org/licenses/by-nc-sa/3.0/

particular set of annotation conventions into another format based on other annotation conventions.

REFERENCES

1. Barbu-Mititelu, V. 2013. *Derivational Semantic Network for Romanian*, Bucharest, National Museum of Romanian Literature Press.
2. Călăcean, M. & Nivre, J. 2008. Data-driven Dependency Parsing for Romanian, Acta Universitatis Upsaliensis (pp. 65-76).
3. Cristea, D., Mihăilă, C., Forăscu, C., Trandabăţ, D., Husarciuc, M., Haja, G. & Postolache, O. 2004. Mapping Princeton WordNet Synsets onto Romanian WordNet Synsets. In *Romanian Journal on Information Science and Technology*, *Special Issue on BalkaNet*, 7, 125-145.
4. Cristea, D. & Pistol, I. C. 2008. Managing language resources and tools using a hierarchy of annotation schemas. In proceedings of *the Workshop on Sustainability o f Language Resources, LREC, Marakesh*. (2008)
5. Curteanu, N., Moruz, M., Trandabat, D., Bolea, C. & Dornescu, I. 2006. The structure and parsing of Romanian verbal group and predicate. In proceedings of the *4th European Conference on Intelligent Systems and Technologies, ECIT*.
6. Hall, J., Nivre, J. & Nilsson, J. 2006. Discriminative classifiers for deterministic dependency parsing. In proceedings of the *21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL)*, Main Conference Poster Sessions.
7. Hall, J. & Nilsson, J. 2007. CoNLL-X Shared Task: Multi-lingual Dependency Parsing, MSI report 06060. Växjö University, School of Mathematics and Systems Engineering. (2007)
8. Hristea, F. & Popescu, M. 2003. A dependency grammar approach to syntactic analysis with special reference to Romanian. In F. Hristea & M. Popescu (Eds.), *Building Awareness in Language Technology* (pp. 9-34). Bucharest: University of Bucharest Press.
9. Jezek, E., Magnini, B., Feltracco, A., Bianchini, A. & Popescu, O. 2014. Structures for linguistic analysis and semantic processing. In proceedings of *LREC* (pp. 890-895).
10. Levin, B. 1993. *English Verb Class and Alternations: A Preliminary Investigation*. Chicago: University of Chicago Press.

11. Perez, A.-C. 2014. Linguistic Resources for Natural Language Processing. PhD thesis, Al. I. Cuza University, Iaşi.
12. Pistol, I. C. & Cristea, D. 2009. Managing metadata variability. *Milan: Proceedings of the 6th International Workshop on Natural Language Processing and Cognitive Science - NLPCS 2009* (pp. 111-116).
13. Rosa, R., Mašek, J., Mareček, D., Popel, M., Zeman, D., & Žabokrtský, Z. HamleDT 2.0: Thirty Dependency Treebanks Stanfordized. In proceedings of *LREC*. (2014)
14. Sanguinetti, M., Bosco, C. & Cupi, L. 2014. Exploiting catenae in a parallel Treebank alignment. In proceedings LREC 2014, 1824-1831.
15. Trandabăţ, D. 2010. Natural Language Processing Using Semantic Frames, PHD Thesis, Computer Science Faculty, Alexandru Ioan Cuza, University of Iasi, Romania. Available online: <http://students.info.uaic.ro/~dtrandabat/thesis.pdf>.

ACKNOWLEDGEMENTS

Our gratitude goes to computer scientists in natural language processing specialists that we work with and without which our linguistic research would not be possible or would take place today as in the past century, during the decades by listing books in libraries and with more modest results than those now we can get in a few months.

CĂTĂLINA MĂRĂNDUC
FACULTY OF COMPUTER SCIENCE,
ALEXANDRU IOAN CUZA UNIVERSITY OF IASI, ROMANIA.
ACADEMIC INSTITUTE OF LINGUISTICS, BUCHAREST, ROMANIA.
E-MAIL: <CATALINA.MARANDUC@INFO.UAIC.RO>

CENEL AUGUSTO PEREZ
FACULTY OF COMPUTER SCIENCE,
ALEXANDRU IOAN CUZA UNIVERSITY OF IASI, ROMANIA.
E-MAIL: <AUGUSTO.PEREZ@INFO.UAIC.RO>

International Journal of Computational Linguistics and Applications, Vol. 6, No. 2, 2015, pp. 105–116
Received 23/01/2015, Accepted 27/02/2015, Final 09/04/2015.
ISSN 0976-0962, http://ijcla.bahripublications.com

Simple, Fast Semantic Parsing with a Tensor Kernel

DAOUD CLARKE
University of Sussex, Falmer, Brighton, UK

ABSTRACT

We describe a simple approach to semantic parsing based on a tensor product kernel. We extract two feature vectors: one for the query and one for each candidate logical form. We then train a clasifier using the tensor product of the two vectors. Using very simple features for both, our system achieves an average F1 score of 40.1% on the WEBQUESTIONS dataset. This is comparable to more complex systems but is simpler to implement and runs faster.

1. INTRODUCTION

In recent years, the task of semantic parsing for querying large databases has been studied. This task differs from early work in semantic parsing in several ways:

- The databases being queried are typically several orders of magnitude larger, contain much more diverse content, and are less structured.
- In standard semantic parsing approaches, the aim is to learn a logical form to represent a query. In recent approaches the goal is to find the correct answer (entity or set of entities in the database), with learning a logical form a potential byproduct.
- Because of this, the datasets, which would have consisted of queries together with their corresponding logical forms, now

may consist of the queries together with the desired correct answer
- The datasets themselves are much larger, and cover a more diverse range of entities, however there may be a lot of overlap in the type of queries in the dataset.

We believe it is the last of these points that means that simple techniques such as the one we present can work surprisingly well. For example, the WEBQUESTIONS dataset contains 83 questions containing the term "currency"; of these 79 are asking what the currency of a particular country is. These 79 questions can be answered using the same logical form template, thus a system only has to see the term "currency", and identify the correct country in the question to have a very good chance of getting the answer correct.

Knowing this on its own is not enough to build an effective system however. We still need to be able to somehow identify that it is this particular term in the query that is associated with this logical form. In this paper we demonstrate one way that this can be achieved. We build on the paraphrasing approach of [1] in that we use a fixed set of templates to generate a set of candidate logical forms to answer a given query and map each logical form to a natural language expression, its *canonical utterance*. Instead of using a complex paraphrasing model however, we use tensor kernels to find relationships between terms occuring in the query and in the canonical utterance. The virtue of our approach is in its simplicity, which both aids implementation and speeds up execution.

2. BACKGROUND

The task of semantic parsing initially focussed on fairly small problems, such as the GeoQuery dataset, which initially consisted of 250 queries [2] and was later extended to around 1000 queries [3]. Approaches to this task included inductive logic programming [2, 3], probabilistic grammar induction [4, 5],

synchronous grammars [6] and induction of latent logical forms [7], the current state of the art on this type of dataset.

More recently, attention has focussed on answering queries in much larger domains, such as Freebase [8], which contains at the time of writing of around 2.7 billion facts. There are two datasets of queries for this database: FREE917 consisting of 917 questions annotated with logical forms [9], and WEBQUESTIONS which consists of 5,810 question-answer pairs, with no logical forms [10]. Approaches to this task include schema matching [9], inducing latent logical forms [10], application of paraphrasing techniques [1, 11], information extraction [12], learning low dimensional embeddings of words and knowledge base constituents [13] and application of logical reasoning in conjunction with statistical techniques [11]. Note that most of these approaches do not require annotated logical forms, and either induce logical forms when training using the given answers, or bypass them altogether.

2.1. *Semantic parsing via paraphrasing*

The PARASEMPRE system of [1] is based on the idea of generating a set of candidate logical forms from the query using a set of templates. For example, the query *Who did Brad Pitt play in Troy?* would generate the logical form

```
Character.(Actor.BraddPitt ⊓ Film.Troy)
```

as well as many incorrect logical forms. These are built by finding substrings of the query that approximately match Freebase entities and then applying relations that match the type of the entity. Given a logical form, a canonical utterance is generated, again using a set of rules, which depend on the syntactic type of the description of the entities.

To identify the most likely logical form given a query, a set of features are extracted from the query, logical form and canonical utterance:

```
what caused the asian currency crisis?
what countries use the euro as official
      currency?
what currency can you use in aruba?
what currency do i bring to cuba?
what currency do i need in cuba?
what currency do i need in egypt?
what currency do i take to turkey?
what currency do italy have?
what currency do mexico use?
what currency do the ukraine use?
what currency do they accept in kenya?
what currency do they use in qatar?
what currency do you use in costa rica?
what currency does brazil use?
what currency does greece use 2012?
what currency does greece use?
what currency does hungary have?
what currency does jamaica accept?
what currency does ontario canada use?
what currency does senegal use?
what currency does south africa have?
what currency does thailand accept?
what currency does thailand use?
what currency does the dominican republic?
what currency does turkey accept?
what currency in dominican republic should i
      bring?
what currency is best to take to dominican
      republic?
what currency is used in england 2012?
what currency is used in france before euro?
what currency is used in germany 2012?
what currency is used in hungary?
what currency is used in switzerland 2012?
what currency should i bring to italy?
what currency should i take to dubai?
what currency should i take to jamaica?
what currency should i take to mauritius?
what currency should you take to thailand?
what currency to take to side turkey?
what do you call russian currency?
what is australian currency?
what is currency in dominican republic?
what is currency in panama?
what is the best currency to take to egypt
      2013?
what is the currency in australia 2011?
what is the currency in croatia 2012?
what is the currency in england 2012?
what is the currency in france?
what is the currency in germany in 2010?
what is the currency in slovakia 2012?
what is the currency in the dominican republic
       2010?
what is the currency in the dominican republic
       called?
what is the currency in the republic of congo?
what is the currency name of brazil?
what is the currency name of china?
what is the currency of germany in 2010?
what is the currency of mexico called?
what is the currency of spain called?
what is the currency of sweden called?
what is the currency used in brazil?
what is the currency used in tunisia?
what is the local currency in the dominican
      republic?
what is the money currency in guatemala?
what is the money currency in italy?
what is the money currency in switzerland?
what is the name of currency used in spain?
what is the official currency in france?
what kind of currency do they use in thailand?
what kind of currency does cuba use?
what kind of currency does greece have?
what kind of currency does jamaica use?
what kind of currency to bring to mexico?
what money currency does canada use?
what the currency in argentina?
what type of currency does brazil use?
what type of currency does egypt have?
what type of currency does the us have?
what type of currency is used in puerto rico?
what type of currency is used in the united
      kingdom?
what type of currency should i take to mexico?
what's sweden's currency?
what's the egyptian currency?
which country has adopted the euro as its
      currency ( 1 point )?
which country uses euro as its main currency?
```

Figure 1. *Questions from the WEBQUESTIONS dataset containing the term "currency"*

- Features extracted from the logical form itself, such as the size of the denotation of a logical form, i.e. the number of results returned when evaluating the logical form as a query on the database. This is important, since many incorrect logical forms have denotation zero; this feature acts as a filter removing these.
- Features derived from an association model. This involves examining spans in the query and canonical utterance and looking for paraphrases between these spans. These

paraphrases are derived from a large paraphrase corpus and WordNet [14].
- Features derived from a vector space model built using Word2Vec [15].

In an analysis on the development set of WEBQUESTIONS, the authors showed that removing the vector space model lead to a small drop in performance, removing the asssociation model gave a larger drop, and removing both of these halved the performance score.

3. TENSOR KERNERLS FOR SEMANTIC PARSING

We know that simple patterns or occurrences in the query can be used to identify a correct logical form with high probability, as with the "currency" example. We still need some way of identifying these patterns and linking them up to appropriate logical forms. In this section we discuss one approach for doing this.

Our goal is to learn a mapping from queries to logical forms. One way of doing this to consider a fixed number of logical forms for each query sentence, and train a classifier to choose the best logical form given a sentence [1]. In order to use this approach, we need a single feature vector for each pair of queries and logical forms. Our proposal is to extract features for each query and logical form indepdendently, and to take their tensor product as the combined vector. Explicitly, let Q be the set of all possible queries and Λ be the set of all possible logical forms. For each query $q \in Q$ and logical form $\lambda \in \Lambda$ we represent the pair (q, λ) by the vector:

$$\phi(q, \lambda) = \phi_Q(q) \otimes \phi_\Lambda(\lambda)$$

where ϕ_Q and ϕ_Λ map queries and logical forms to a vector space, i.e. perform feature extraction.

Whilst this could potentially be a large space, note that we can use the kernel trick to avoid computing very large vectors, using a simple identity of dot products on tensor spaces:

$$\phi(q_1, \lambda_1) \cdot \phi(q_2, \lambda_2) = (\phi_Q(q_1) \cdot \phi_Q(q_2))\ (\phi_\Lambda(\lambda_1) \cdot \phi_\Lambda(\lambda_2))$$

The advantage of using the tensor product is that it preserves all the information of the original vectors, allowing us to learn how features relating to queries map to features relating to logical forms.

More generally, instead of representing the query and logical form as vectors directly, this can be done implicitly using kernels. For example, we may use a
string kernel κ_1 on Q and a tree kernel κ_2 on Λ, then define the kernel $\kappa(q,\lambda) = \kappa_1(q)\ \kappa_2(\lambda)$ on $Q \times \Lambda$. This idea is closely related to the Schur product kernel [16].

It is worth noting at this point that, while what we really want is a one-to-one mapping from queries to logical forms, the classifier actually gives us a set of logical forms for each query: we simply ask it to classify each pair (q, λ). In a probabilistic approach, such as logistic regression, we can choose the λ for which the classifier gives the highest probability for (q, λ).

3.1. *Application to semantic parsing via paraphrasing*

There are clearly many ways we could map queries and logical forms to vectors. In this paper we will consider one simple approach in which we use unigrams as the features for both the query and the canonical utterance associated with the logical form. In this case, the tensor product of the vectors corresponds directly to the cartesian product of the unigrams derived from the query with those from the canonical utterance.

Recall that given two vector spaces U and V of dimensionality n and m, the tensor product space $U \otimes V$ has dimensionality nm. If we have bases for U and V, then we can construct a basis for $U \otimes V$. For each pair of basis vectors u and v in U and V respectively, we take a single basis vector $u \otimes v \in$

$U \otimes V$. In our case, the dimensions of U and V correspond to terms that can occur as unigram features in the query or canonical utterance respectively. Thus each basis vector of $U \otimes V$ corresponds to a pair of unigram features.

As an example from the WEBQUESTIONS dataset, consider the query, *What 5 countries border ethiopia?*, and the canonical utterance *The adjoins of ethiopia?*, whose associated logical form gives the correct answer. Then there will be a dimension in the tensor product for each pair of words; for example the dimensions associated with (*countries*, *adjoins*) and (*border*, *adjoins*), as well as less useful pairs such as (*5*, *ethiopia*) would all have non-zero values in the tensor product. Thus we are able to learn that if we see borders in the query, then a logical form whose canonical utterance contains the term adjoins is a likely candidate to answer the query.

4. EMPIRICAL EVALUATION

4.1. *Dataset*

We evaluated our system on the WEBQUESTIONS dataset [10]. This consists of 5,810 question-answer pairs. The questions were obtained by querying the Google Suggest API, and answers were obtained using Amazon Mechanical Turk. We used the standard train/test split supplied with the dataset, and used cross-validation on the training set for development purposes.

4.2. *Implementation*

We built our implementation on top of the PARASEMPRE system [1], and so our evaluation exactly matches theirs. Our implementation is freely available online.[1] We substituted the paraphrase system of PARASEMPRE with our tensor kernel-based system (i.e. we excluded features from both the association and vector space models), but we included the PARASEMPRE features derived from logical forms.

[1] Location witheld to preserve anonymity

To implement our tensor kernel of unigram features, we simply added all pairs of terms in the query and canonical utterance as features; in preliminary experiments we found that this was fast enough and we did not need to use the kernel trick, which could potentially provide further speed-ups. We did not implement any feature selection methods which may also help with efficiency.

For evaluation, we report the average of the F1 score measured on the set of entities returned by the logical form when evaluated on the database, when compared to the correct set of entities. This allows, for example, to get a non-zero score for returning a similar set of entities to the correct one. For example, if we return the set {Jaxon Bieber} as an answer to the query *Who is Justin Bieber's brother?* we allow a nonzero score (the correct answer according to the dataset is {Jazmyn Bieber, Jaxon Bieberg).

4.3. *Results*

Results are reported in Table 1. Our system achieves an average F1 score of 40.1%, compared to PARASEMPRE'S 39.9%. Our system runs faster however, due to the simpler method of generating features. Evaluating using PARASEMPRE on the development set took 22h31m; using the tensor kernel took 14h44m on a comparable machine.

Since we have adopted the logical form templates of PARASEMPRE, our upper bound or oracle F1 score is the same, 63% [1]. This is the score that would be obtained if we knew which was the best logical form out of all those generated. In contrast, Microsoft's DEEPQA has an oracle F1 score of 77.3% [11]; this could account for a large amount of the overall increase in their system. There is no reported oracle score for the Facebook system [13].

5. DISCUSSION

Table 2 shows the top unigram feature pairs after training on the WEBQUESTIONS training set. It is clear that, whilst there are some superfluous features that simply learn to replace a word

with itself (for example *currency* with *currency*, there are obviously many useful features that would be nontrivial to identify accurately. There are also spurious ones such as the pair (*live*, *birthplace*); this is perhaps due to a large proportion of people who live in their birthplace.

Table 1. *Results on the WEBQUESTIONS dataset, together with results reported in the literature*

	Average F1 score
SEMPRE [10]	35.7
PARASEMPRE [1]	39.9
FACEBOOK [13]	41.8
DEEPQA [11]	45.3
Tensor kernel with unigrams	40.1

Table 2. *Top unigram pair features and their weights after training*

Feature	Weight	Feature	Weight
(currency, currency)	4.18	(name, who)	2.69
(parents, father)	3.46	(born, birth)	2.69
(die, death)	3.33	(influenced, influenced)	2.64
(religion, religion)	3.28	(live, birthplace)	2.63
(currency, used)	3.22	(country, birthplace)	2.62
(religions, religion)	3.11	(type, form)	2.62
(movies, film)	2.97	(do, profession)	2.60
(states, adjoins)	2.97	(died, death)	2.60
(timezone, zone)	2.95	(system, form)	2.60
(timezone, time)	2.94	(countries, country)	2.60
(speak, spoken)	2.91	(married, marry)	2.55
(currency, countries)	2.84	(language, language)	2.54
(money, currency)	2.82	(music, genres)	2.51
(capital, city)	2.77	(money, used)	2.47
(party, party)	2.75	(time, zone)	2.47
(nationality, country)	2.72	(wife, spouse)	2.46

In development, we found that ordering the training alphabetically by the text of the query lead to a large reduction in accuracy.[2] Ordering alphabetically when performing the split for

[2] We omit the values since they were performed on an earlier version of our code and are not comparable.

cross validation (instead of random ordering) means that a lot of queries on the same topic are grouped together, increasing the likelihood that a query on a topic seen at test time would not have been seen at training time. This validates our hypothesis that simple techniques work well because of the homogeneous nature of the dataset. We would argue that this does not invalidate the techniques however, as it is likely that real-world datasets also have this property.

It is a feature of our tensor product model that there is no direct interaction between the features from the query and those from the logical form. This is evidenced by the fact that the system has to *learn* that the term $currency$ in the query maps to $currency$ in the canonical utterance. This hints at ways of improving over our current system. More interestingly, it also means that we are currently making very light use of the canonical utterance generation; in the canonical utterance, $currency$ could be replaced by any symbol and our system would learn the same relationship. This points at another route of investigation involving generating features for use in the tensor kernel directly from the logical form instead of via canonical utterances.

6. CONCLUSION

We have shown semantic parsing via paraphrasing using unigram features together with a tensor kernel performs comparably to more complex systems on the WEBQUESTIONS dataset. Our system is simpler to implement and runs faster.

In future work, as well as looking at more sophisticated feature inputs to the tensor kernel, we hope to work on improving the oracle F1 score.

REFERENCES

1. Berant, J. & Liang, P. 2014. Semantic parsing via paraphrasing. In proceedings of the *52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Baltimore,

Maryland, Association for Computational Linguistics (pp. 1415-1425).
2. Zelle, J. M. & Mooney, R. J. 1996. Learning to parse database queries using inductive logic programming. In proceedings of the *National Conference on Artificial Intelligence* (pp. 1050-1055).
3. Tang, L. R. & Mooney, R. J. 2001. Using multiple clause constructors in inductive logic programming for semantic parsing. In *Machine Learning: ECML 2001* (pp. 466-477), Springer.
4. Zettlemoyer, L. S. & Collins, M. 2005. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In proceedings of the *Twenty-First Conference on Uncertainty in Artificial Intelligence (UAI2005)* (pp. 658-666), AUAI Press.
5. Kwiatkowksi, T., Zettlemoyer, L., Goldwater, S. & Steedman, M. 2010. Inducing probabilistic CCG grammars from logical form with higher-order unification. In proceedings of the *2010 Conference on Empirical Methods in Natural Language Processing, Cambridge* (pp. 1223-1233), MA, Association for Computational Linguistics.
6. Wong, Y.W. & Mooney, R. 2007. Learning synchronous grammars for semantic parsing with lambda calculus. In proceedings of the *45th Annual Meeting of the Association of Computational Linguistics* (pp. 960-967), Prague, Czech Republic, Association for Computational Linguistics.
7. Liang, P., Jordan, M. & Klein, D. 2011. Learning dependency-based compositional semantics. In proceedings of the *49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* (pp. 590-599), Portland, Oregon, USA, Association for Computational Linguistics.
8. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In proceedings of the *2008 ACM SIGMOD International Conference on Management of Data* (pp. 1247-1250), ACM.
9. Cai, Q. & Yates, A. 2013. Large-scale semantic parsing via schema matching and lexicon extension. In proceedings of the *51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 423-433), Sofia, Bulgaria, Association for Computational Linguistics.
10. Berant, J., Chou, A., Frostig, R. & Liang, P. Semantic parsing on Freebase from question-answer pairs. In proceedings of the *2013*

Conference on Empirical Methods in Natural Language Processing (pp. 1533-1544), Seattle, Washington, USA, Association for Computational Linguistics.
11. Wang, Z., Yan, S., Wang, H. & Huang, X. 2014. An Overview of Microsoft Deep QA System on Stanford WebQuestions Benchmark. Technical Report MSR-TR-2014-121.
12. Yao, X., Berant, J. & Durme, B. V. 2014. Freebase qa: Information extraction or semantic parsing? In *Workshop on Semantic Parsing*.
13. Bordes, A., Chopra, S. & Weston, J. 2014. Question answering with subgraph embeddings. In proceedings of the *2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 615-620), Doha, Qatar, Association for Computational Linguistics.
14. Fellbaum, C. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.
15. Mikolov, T., Chen, K., Corrado, G. & Dean, J. 2013. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.
16. Shawe-Taylor, J. & Cristianini, N. 2004. Kernel methods for pattern analysis. Cambridge University Press, pub-CAMBRIDGE:adr.

DAOUD CLARKE
DEPARTMENT OF INFORMATICS,
UNIVERSITY OF SUSSEX,
FALMER, BRIGHTON, UK.
E-MAIL: <DAOUD.CLARKE@GMAIL.COM>

International Journal of Computational Linguistics and Applications, Vol. 6, No. 2, 2015, pp. 117–140
Received 25/01/2015, Accepted 25/02/2015, Final 04/03/2015.
ISSN 0976-0962, http://ijcla.bahripublications.com

Cross-lingual Semantic Generalization for the Detection of Metaphor

MICHAEL MOHLER
MARC TOMLINSON
BRYAN RINK
Language Computer Corporation, Richardson, TX

ABSTRACT

In this work, we describe a supervised cross-lingual methodology for detecting novel and conventionalized metaphors that derives generalized semantic patterns from a collection of metaphor annotations. For this purpose, we model each metaphor annotation as an abstract tuple – (source, target, relation, metaphoricity) – that packages a metaphoricity judgement with a relational grounding of the source and target lexical units in text. From these annotations, we derive a set of semantic patterns using a three-step process. First, we employ several generalized representations of the target using a variety of WordNet information and representative domain terms. Then, we generalize relations using a rule-based, pseudo-semantic role labeling. Finally, we generalize the source by partitioning a semantic hierarchy (defined by the target and the relation) into metaphoric and non-metaphoric regions so as to optimally account for the evidence in the annotated data. Experiments show that by varying the generality of the source, target, and relation representations in our derived patterns, we are able to significantly extend the impact of our annotations, detecting metaphors in a variety of domains at an F-measure of between 0.88 and 0.92 for English, Spanish, Russian, and Farsi. This generalization process both enhances our ability to jointly detect novel and conventionalized metaphors and enables us to transfer the knowledge encoded in metaphoricity annotations to novel languages.

Keywords: Metaphor detection, generalization, semantic modeling, WordNet, transfer learning

1. INTRODUCTION

Metaphor is the air we breathe. It stirs the emotions, arouses the senses, and serves as a vehicle for describing and reasoning about difficult concepts – all while using language that is both clear and familiar. Metaphor is everywhere in human language, hiding in plain sight. For this reason, it is crucial for technologies that seek to model and understand human language to be capable of correctly identifying and interpreting metaphor. Indeed, metaphor has been found to confound both statistical and knowledge-based techniques for natural language processing across a wide variety of applications including textual entailment, text summarization, word sense disambiguation, semantic textual similarity, question answering, and event extraction. In this work, we propose a methodology that derives generalized semantic patterns from existing metaphor annotations in order to detect a wide variety of metaphoric language as either a stand-alone system or as a component in a larger supervised or unsupervised metaphor detection system.

Although there have been many influential theories regarding the cognitive basis of metaphor, the most prominent is Lakoff's Contemporary Theory of Metaphor [13, 11], which popularized the idea of a conceptual metaphor mapping. Within the cognitive framework of a given conceptual mapping, terms pertaining to one concept (the *source*) can be used figuratively to express some aspect of another concept (the *target*). For example, the conceptual metaphor "Life is a Journey" indicates a cognitive lens through which the target concept "life" may be more easily discussed and understood. This particular mapping allows us to speak of one being stuck in a "dead-end" job, of a crucial decision as being a "fork in the road", and of someone's life "taking a wrong turn".

Existing work on the identification of metaphor can be broadly categorized as either feature-based or example-based. Feature-based metaphor identification is based upon the

assumption that metaphoric usages in context typically have certain characteristics that serve as cues for indicating non-literal usage. Such characteristic indicators of non-literal usage include the pairing of abstract with concrete terms [28, 4, 3, 27], the violation of selectional preference [7, 19, 10, 29], dissonance between a term and its greater context [26, 1, 25], explicit linguistic cues [16], and semantic unrelatedness between terms [22, 5]. In general, such methods are successful at detecting novel metaphors, but have difficulty in detecting commonly used figurative language or "conventionalized metaphors" for which modeling selectional preference, contextual relatedness, and semantic mismatch is more complex.

Example-based methods seek to detect metaphor by comparing candidate texts to a set of known metaphors using abstraction hierarchies [18], known conceptual metaphor domain interactions [21], semantic signatures [20], models of metaphoricity priors [24], probabilistic typicality hierarchies [15], or simply large lexical stores of conventionalized Metaphor [14]. Of course, methods that compare candidates to known metaphors are able to reliably detect those that are most commonly used, but they require some additional framework to generalize from these examples to less common or even novel metaphoric utterances.

We propose an approach to metaphor identification that follows the example-based paradigm but differs from existing work in that we (1) explicitly explore a variety of methods for generalization; (2) determine the impact that such methods have on overall metaphor identification performance; and (3) apply these examples cross-lingually to maximize performance in novel languages. In particular, we generalize our annotations, which consist of the tuple (source, target, relation, metaphoricity), in three ways:

1. We generalize target lexemes using semantic categories and a domain-level groupings of terms and exploit the dichotomy between abstract and concrete nouns;

2. We generalize source/target relations in text by converting dependency relations into broader pseudo-semantic relations using a rule-based approach;
3. We generalize source lexemes by exploiting the hypernymy links in a semantic hierarchy (i.e., WordNet).

For each generalized target/relation pair, we map the associated source lexemes from our annotations onto the semantic hierarchy. In particular, we make use of a largescale, multilingual semantic knowledge base (i.e., a multilingual WordNet) that combines groupings of related objects (i.e., synsets and semantic categories) with a hierarchical tree structure (i.e., hypernymy relations). For a given target/relation pair, we then define a semantic pattern as a single node in this hierarchy with an associated metaphoricity judgement which indicates that that node, and all of its descendents in the hierarchy, are or are not metaphors unless they are under the influence of a different pattern node. In other words, the metaphoricity judgement of each node in the hierarchy is determined by the metaphoricity associated with the pattern node that is its nearest ancestor. The set of pattern nodes is selected using a dynamic programming algorithm to optimally select (and assign judgements to) a minimal set of nodes in the hierarchy so as to account for all of the annotation evidence. In effect, we are using a multilingual knowledge base to generalize from the given examples (of both literal and metaphoric usages) so as to partition the semantic space that it defines into regions of likely metaphoricity and regions of unlikely metaphoricity. By generalizing in this way, we are able to transfer knowledge of metaphor to languages with no metaphor annotations.

The remainder of this work is organized as follows. In Section 2, we survey related work in metaphor identification. Then, in Section 3, we describe the two components of our system – generalizing through the knowledge base to produce a set of semantic patterns and using the resulting patterns to detect metaphor in unseen data. Section 4 describes the provenance and the characteristics of the multilingual datasets we use to train and evaluate our system. In Section 5 we present our experiments and

discuss our results. Finally, we share the insights gained from these experiments in Section 6 and offer our recommendations for moving forward.

2. RELATED WORK

One of the earliest works in example-based metaphor processing is that of Martin [17], who sought to enable an automated Unix help client (uc) to detect and interpret conventional metaphor. This system had as its backing an abstraction hierarchy which was used to enable the interpretation components of the system to generalize. In particular, it would attempt to apply manually-coded interpretations associated with a particular type of metaphor, and then, if it failed, would move away from the original metaphor to more and more abstract representations until the metaphor could be understood. Originally, the system was backed by only twenty-two core metaphors (with interpretations), but these were further expanded to 200 as part of the Berkeley Master Metaphor List [12]. In many ways, we follow the intuitions of this work on a much larger scale, using a wide variety of metaphoricity annotations across multiple unrelated domains in multiple languages.

Krishnakumaran and Zhu [10] introduced a key observation – that metaphors can be categorized according to their relationship to some non-metaphoric unit in the text and that the characteristics that indicate metaphoricity differ by category. They proposed three types of metaphors – IS-A metaphors (Type I), verb-noun metaphors (Type II), and adj-noun metaphors (Type III). In order to detect these three types of metaphors, they made extensive use of the WordNet hypernym structure (to rule out literal IS-A relations) and a verb-adj/noun co-occurrence matrix (to rule out common pairings). In effect, they employed co-occurrence information to partition the knowledge base into regions of conventional usage versus unconventional usage which is, arguably, an approximation of (novel) metaphorical usage. More recently, Li et al. [15] built upon this idea by combining extracted figurative comparisons with a probabilistic IS-A knowledge base (ProBase) to partition the semantic space

associated with particular dependency relations in a much more principled way.

Likewise, Hovy et al. [9] focused on the semantic relation between pairs of words in a text by building a tree-kernel classifier that uses (1) a vector representation of individual words and (2) several tree representations of the word. These representations included lemmatized versions of the words, POS tags of the words, and WordNet semantic classes (i.e., lexicographer files). This work represents a clear attempt to generalize from the surface form of the words in their training data. However, by using the full parse tree instead of a relation between the source and target, their methodology was particularly vulnerable to data sparsity.

Similar to our approach in rationale, if not methodology, is Mohler et al. [20], which compared sentence-level utterances against a large collection of sentences validated as either containing metaphors or not. In particular, they sought to compare sentences within a semantic space (defined by a WordNet- and Wikipedia-based "semantic signature"). While this approach successfully addressed the semantics of metaphor, it did not consider the relationship between the source and the target within a sentence. As such, it was heavily impacted by noise associated with the wider context of the sentence.

At the forefront of large-scale, example-based metaphor detection is the work of Levin et al. [14], who have produced a resource containing common metaphors associated with a variety of target concepts in English, Spanish, Russian, and Farsi. Moving beyond the relational categories of Krishnakumaran and Zhu [10], they predicted metaphoricity between lexical pairs in a variety of dependency relations: subj-verb, obj-verb, adv-verb, adj-noun, noun-pred, noun-noun, noun-poss, and noun-prep-noun. The most common terms that co-occur with a target term (e.g., "poverty", "wealth", "taxation") were manually analyzed and annotated as being either conventionalized metaphors or literal language. However, other than normalizing for conjunctions and several semantically "light" nouns (e.g., containers, quantifiers, partitives), no generalization was carried out. Without employing generalization, it is not possible to detect

novel metaphors using a resource such as theirs.

3. METHODOLOGY

We propose a supervised approach to the identification of metaphor which seeks to generalize over an existing dataset annotated for metaphoricity. We model each annotation (without contextual information) using the abstract tuple – (source, target, relation, metaphoricity). It is our hypothesis that the vast majority of utterances in text, represented as such a tuple, are either consistently metaphoric or consistently nonmetaphoric, with the wider context of the utterance having little to no effect on this property.[1] Building on this hypothesis, we assert that a model of the prior metaphorical likelihood of all possible utterances – that is, all possible source/target pairs in all dependency relations[2] – represents a complete solution to the metaphoricity problem. We seek to approximate this level of knowledge by deriving and using semantic patterns that are capable of grouping the metaphoricity decisions of individual examples and applying them to a bounded region of this tuple space. These decisions can then be propagated to utterances in larger and more general regions without the need for humans to annotate such (potentially novel) utterances directly.

While this approach was developed as a supplement to an existing feature-based metaphor identification system [3], we have also made use of it as a stand-alone system, and we employ

---

[1] That said, we acknowledge several classes of utterances such as "Cholera is a disease of poverty"; "Men are animals"; and "The rock began to sing (in a dream)" for which this hypothesis is insufficient. However, we believe that the appropriate means for handling such cases is to determine a metaphoric or non-metaphoric prior likelihood for the utterance and to overcome this prior in anomalous cases using supplementary, context-dependent components tailored to such cases.

[2] This is defined by the set of tuples $(S^V,\ T^V,\ R^N,\ m \in [0..1])$ where V is the vocabulary size and N is the number of possible syntactic elations between two words within a sentence.

it as such throughout this work. Our approach consists of two stages – (1) generalizing our annotations into semantic patterns using a three-step process; and (2) using these semantic patterns to detect linguistic metaphors in unseen data. In this section, we describe these two components with a particular focus on our distinct techniques for individually generalizing the target, the relation, and the source. We then describe our method for combining the results yielded by these patterns to arrive at a single decision for a given input.

3.1. *Generalizing over existing annotations*

In order to effectively generalize from our metaphoricity annotations, we first individually generalize the target and the relation. For the target, we make use of a variety of semantic information associated with its representation in WordNet including (1) its associated synset, (2) its semantic category, and (3) whether it can be considered "concrete" or "abstract". In addition, we generalize the target using a manual grouping of terms related to a small set of target domains. The domains under consideration – GOVERNMENT, BUREAUCRACY, DEMOCRACY, ELECTIONS, POVERTY, WEALTH, and TAXATION – were selected to represent a variety of distinct domains with different characteristics.[3] Relations are generalized using a rule-based, pseudo-semantic role labeling process which maps from dependency chains to a small set of abstract semantic relations.

For each target/relation representation pair, we define a WordNet semantic hierarchy upon which we can map individual source lexemes, along with their metaphoricity annotations. Once these annotations have been linked to WordNet, it is possible to begin the process of deriving semantic patterns. We define a semantic pattern according to the tuple, $(g(S, x), T, R, m)$, where T corresponds to some representation of the target lexeme, R corresponds to some representation of the relation between a

[3] These domains correspond to those under consideration as part of the IARPA Metaphor program.

source and target within a text, m corresponds to a binary metaphoricity decision (metaphor or non-metaphor), and the function g (S,x) defines a group within the semantic space S (such as all nodes in a subtree) that contains the annotated source x. In the sections to follow, we will describe several techniques for representing T and R at various levels of generality along with our methodology for partitioning the semantic space S (for a given T, R) into regions within which we assert that m must be either always true or always false.

Generalizing Targets In addition to the lexical (surface form) representation of a target, T_{LEX}, we propose four alternative representations. First, we link the target lexeme to the WordNet synset that corresponds to its most frequent sense[4] – T_{SYN}. This enables us to directly propagate metaphoricity annotations to all synonyms of a given term. More importantly, this allows us to propagate annotations to synonyms cross-lingually using the cross-lingual links (within a synset) associated with our multi-lingual version of WordNet.

Next, we represent a given target using the WordNet semantic category (i.e., lexicographer file) associated with its most frequent sense – T_{SEM}. This permits us to infer the metaphoricity of the tuple (devours, NOUN.STATE, dobj) from the annotated example (devours, poverty, dobj). This is because the semantic category NOUN.STATE includes additional target lexemes "health", "silence", "guilt", and "comfort" – none of which is literally capable of "devouring".

Third, we partition the noun hierarchy of WordNet into an abstract-concrete dichotomy – T_{ABS} – which indicates whether the synset PHYSICAL ENTITY is or is not a direct or indirect hypernym of the target. While this generalizes the targets in a very coarse way, a significant number of physical interaction

[4] This was determined to be sufficient for the limited number of target lexemes that we consider, but will need to be reconsidered moving forward.

verbs (e.g., “drop”, “carry”, “throw”, “touch”, “eat”) will be metaphoric for all abstract nouns.

Finally, we represent the target according to the domain or topic with which it is associated – T_{DOM}. For our purposes, this corresponds to a manually-created list of lexical items, each related to one of the seven domains mentioned above and separated according to part-of-speech. This permits us to group together such target lexemes as “taxes”, “taxation”, and “income tax”.

3.2. *Generalizing relations*

For representing the relation, R, which links the source and the target in text, we explore two possibilities. First, we use the dependency relation (as determined by MaltParser) between the source and the target directly – R_{DEP}. This relation is then post-processed to remove conjunction relations (“conj”), so that both “government” and “bureaucracy” in the phrase “government and bureaucracy punish us” will have the same subject relation to the verb “punish”.

In order to better promote generalization, we also represent the relation, R, by transforming it into a language-independent, pseudo-semantic relation – R_{SEM} – derived using a small number of manually-crafted rules over raw dependency relations. For our purposes, we define a small set of pseudo-semantic relations – AGENCY, PATIENCY, and MODIFICATION[5] between the source and target. As an example, each of the following phrases associates the target (“wealth”) as an AGENT to some form of the source verb “enslave”: “wealth enslaves people” (nsubj), “wealth continues to enslave people” (nsubj+xcomp^{-1}), “wealth which enslaves people” (rcmod^{-1}), “causes wealth to enslave people” (infmod^{-1}), “for wealth to enslave people” (prep for+infmod^{-1}), and “chained to my enslaving wealth” (amod^{-1}). Because in each of these cases “wealth” is metaphorically “enslaving” someone, we avoid sparsity (and improve coverage) by propagating our annotations to all of these relations through

[5] We are concerned here with adjectival modifiers only.

this generalization step. We define equivalent rules to transform raw dependency relations to these three pseudosemantic relations for each of our four languages.[6] Since these pseudo-semantic relations are consistent across languages, their use enables us to propagate annotations across languages by pairing these relations with any of the language independent target representations (i.e., T_{SYN}, T_{ABS}, T_{SEM}, or T_{DOM}) and deriving cross-lingual semantic patterns for the pair's associated hierarchy.

Table 1. *Candidate LM sources – categorized according to metaphoricity – for "bureaucratic [SOURCE]" given the adjective-noun dependency relation (amod)*

Metaphoric	Candidate	LM Sources	Non-Metaphoric	Candidate	LM Sources
Maze	Monster	Fiefdoms	Process	Management	Inconsistencies
Nightmare	Sorcery	System	Control	Activity	Administration
Hell	Basement	Perdition	Adjudication	Tenure	Occupation

Generalizing Sources Throughout the Knowledge Base Once we have defined a target/ relation pair, we begin the process of generalizing our source lexemes and deriving semantic patterns. Table 1 shows a variety of source lexical items taken from our annotations that share an "amod^{-1}" relation with the target lexeme "bureaucratic"[7] Before generalizing, we must first link each of these annotated source lexemes into a semantic knowledge base, such as WordNet[8]. WordNet represents an ideal knowledge base for our purposes due to its ability to simultaneously group individual senses (i.e., synsets and semantic categories) and to define a hierarchical structure within groups using hypernymy relations. However, the problem of

[6] While these rule-based groupings do not approach the accuracy of state-of-the-art semantic role labeling (SRL) systems (at least in English), we believe that their quality is sufficient for the task of metaphor identification. The use of move advanced SRL technologies for this task remains an open problem.

[7] These are organized as "metaphoric" or "non-metaphoric" according to the predominant metaphoricity judgement in our annotations.

word sense ambiguity makes it non-trivial to link individual lexical items to particular synsets. This is even more problematic when the lexical items to be linked are being used in non-literal ways. To account for this, we perform a light disambiguation step that filters out potential senses whose annotated metaphoricity neighborhood (in the semantic hierarchy) fails to match the metaphoricity of the annotation. For example, the lexeme "Hell" can be linked to either a synset that contains the word "Perdition" or a synset with a hypernym of "Mischief". Given the existence of an annotation for "Perdition" with the same metaphoricity, the first synset is preferred. After limiting the number of potential word-sense mappings in this way[8], annotated source terms are linked to all remaining senses.

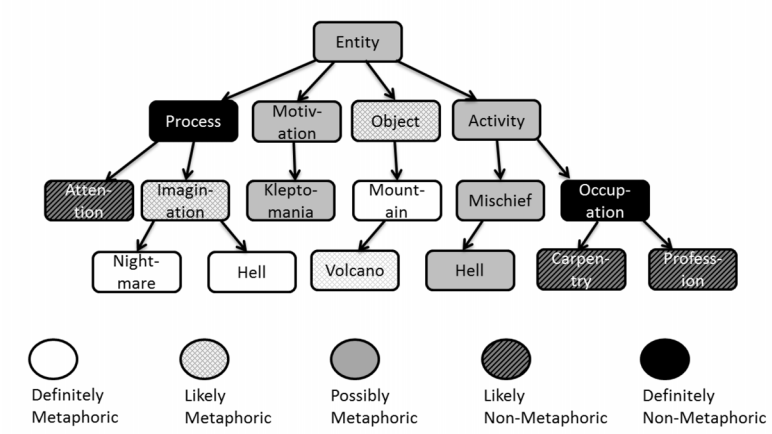


Figure 1. *Categorizing nodes in the source semantic hierarchy based upon the annotated source lexemes from Table 1.*

After each annotation has been linked to one or more nodes in the source semantic hierarchy, metaphoricity decisions are propagated to the nodes themselves. Each node in the hierarchy is categorized as one of the following:

[8] Further details omitted, due to space.

1. **definitely non-metaphoric** – if the annotations for a particular node are entirely negative;
2. **definitely metaphoric** – if the annotations for the node are at least partially positive;
3. **likely non-metaphoric** – if there are no direct annotations, but all of the node's direct or indirect ancestors or descendants are known to be non-metaphoric;
4. **likely metaphoric** – if there are no direct annotations, but all of the node's direct or indirect ancestors or descendants are known to be metaphoric;
5. **possibly metaphoric** – if there is no indication at all among the existing annotations (or if the results are mixed).

The associated categorization of each node in a simplified source hierarchy (based on the annotations from Table 1) is shown in Figure 1.

At this point, we must define our semantic patterns by selecting a set of nodes in the source hierarchy (and assign a metaphoricity judgement to each) such that the metaphoricity judgement associated with each node's closest pattern-node ancestor will determine whether it is likely to be metaphoric or not. We constrain our pattern selection process such that it is constrained to correctly categorize all "definitely metaphoric" and "definitely non-metaphoric" nodes (i.e., those representing the annotations themselves). The remaining categories ("likely metaphoric", "possibly metaphoric", and "likely nonmetaphoric") represent a continuum along which the system can increase recall at the expense of precision. For tasks that require a higher recall, the system can be further constrained to select, for instance, "likely metaphoric" nodes as metaphors as well. Any nodes in groups that have not been constrained in this way may be dominated by patterns suggesting metaphoricity or non-metaphoricity or they may be dominated by no pattern at all.

In order to encourage generalization throughout the source hierarchy, we wish to select a minimal set of patterns nodes that satisfy our constraints. These nodes can be selected efficiently using a straightforward, tree-based dynamic programming methodology in which the state space is defined by two

dimensions: (1) the current node, *N*, and (2) the current metaphoricity judgment, *m*, defined by the pattern that dominates it (i.e., the nearest pattern node on its ancestor chain). If the system has been constrained to judge the current node as metaphoric or non-metaphoric (as described above), some semantic pattern must be correctly applied to this node – either the pattern that currently dominates the node, or a new pattern defined on the node itself. Otherwise, if the current node is under no constraints, we are free to assign a metaphoricity pattern or a non-metaphoricity pattern to the node or to add no pattern and allow any ancestor pattern in effect to continue dominating the hierarchy. This process is described more formally by the following equations[9] where r(*N*) represents the required constraints on node N – i.e., "metaphor" (met), "non-metaphor" (lit), "unconstrained" (unc) – and N_c represents a direct descendant of N in the source semantic hierarchy:

$$f(N,m) = \begin{cases} \min\limits_{x \in h(m)} g(N,m,x) & \text{if } r(N) = unc \\ g(N,m,r(N)) \end{cases} \quad (1)$$

$$g(N,m,x) = \begin{cases} \sum\limits_{c} f(N_c,x) & \text{if } m = x \\ 1 + \sum\limits_{c} f(N_c,x) \end{cases} \quad (2)$$

$$h(m) = \begin{cases} \{met, lit, unc\} & \text{if } m = unc \\ \{met, lit\} \end{cases} \quad (3)$$

The count of the minimal number of patterns can be computed by summing the results of $f(N_R,$ "unc") for each root node, N_R, in

[9] In effect, $h(m)$ is the set of potential pattern decisions that can be selected based on the ancestor patterns; $g(N, m, x)$ is the number of new patterns required to change the current node *N*, from m to x, and to correctly cover all descendants; and $f(N,m)$ is the number of new patterns required to cover the current node, *N*, and all of its descendants given the ancestor pattern m.

the hierarchy. Once these counts have been computed over the state space, selecting the nodes that lead to this optimal state can be accomplished by greedily traversing the state space along locally optimal paths. Note that the result of this process is a partitioning of the entire semantic space of the source hierarchy into metaphoric, non-metaphoric, and unclear regions for a given target/relation pair. Once a set of pattern nodes has been selected that satisfies our constraints, they can be used either as a stand-alone, semantics-only metaphor detection system or in conjunction with an existing metaphor detection system.

3.3. *Detection of metaphors in unseen text*

Employing these patterns in a stand-alone metaphor detection system requires two things: (1) a strategy for selecting potential source/target pairs in text and (2) an algorithm for combining the metaphoricity decisions associated with multiple target/relation representations. For the first of these, we begin with a list of lexical items that have been manually associated with our seven target domains. These have been supplemented using word senses gathered from target domain signatures for each domain in the manner of Bracewell et al.[2]. For each target term selected, we consider all content words in the same sentence as potential sources.[10]

Once the source/target pairs have been extracted from a text, they are converted into (source, target, relation) tuples – for each target/relation representation – and compared against the patterns derived in Section 3.1. We then treat our patterns as a cascade, analyzing groups in increasing order of abstractness of the target representation – T_{LEX}, T_{SYN}, T_{SEM}, T_{DOM}, T_{ABS}. Within each group, we compare the two relation representations – R_{DEP} and R_{SEM} – which can either provide the same metaphoricity

[10] We additionally collapse hyphenated terms and collocations when selecting both the source and the target. For collocations and hyphenated terms that contain a target term, we produce a sub-collocate candidate pair – e.g., (stricken, poverty, dep) for “poverty-stricken”.

judgement, different metaphoricity judgments, or indicate that there is no clear decision. If they agree (or if only one provides a clear answer), then that metaphoricity decision is assigned to the pair. If they disagree (or cannot provide an answer), the next group is considered. If none of the groups in the cascade results in a response, the metaphoricity of the pair remains unclear and is reported as such. This represents a cascading structure from specific to more general representations of the target and relation.

4. DATASETS

In order to evaluate our methodology for generalizing over metaphor annotations, we make use of four datasets (in four languages – English, Spanish, Russian, and Farsi) developed by a team of annotators with native-level proficiency. The size and characteristics of each dataset are summarized in Table 2. The first dataset (ANN) consists of examples selected by the annotators using targeted web searches for representative (non-conventionalized) metaphors for particular pairs of source and target concepts that were of interest at the program level. While this dataset is a good source of novel metaphors in a variety of source and target domains, it includes only a single annotation (with source and target) for the full sentence and does not include any non-metaphor annotations.

The second dataset (REC) was developed to address this problem by providing a more natural source of data for training the machine learning component of our overall system with a significant number of non-metaphor annotations. Individual documents were selected automatically to be annotated thoroughly. For these documents, annotators were provided with all source/target pairs that had been selected as described in Section 3.3. This is the most 'natural' of our four datasets. The third (EVAL) was annotated in the same way by our annotators, but was provided by a third party for the purpose of evaluating our system's ability to detect metaphors in unseen data. As such, it has a slight bias towards metaphoricity.

Table 2. *Datasets used in our experiments with size and balance information. The REDUCED set corresponds to the subset of the combined dataset that can be represented using a pseudosemantic relation representation as described in Section 3.2.*

Language	Dataset	Metaphors	Non-Metaphors	Unclear	Language	Dataset	Metaphors	Non-Metaphors	Unclear
	ANN	8,667	0	125		ANN	4,308	0	353
	REC	402	25,238	340		REC	223	43,241	1,213
EN	EVAL	284	7,130	177	ES	EVAL	177	15,533	400
	SYS	8,507	29,563	6,562		SYS	7,053	28,782	8,633
	TOTAL	17,860	61,931	7,204		TOTAL	11,761	87,556	10,599
	REDUCED	5,512	7,080	2,683		REDUCED	5,158	10,946	4,745
	ANN	3,436	0	322		ANN	2,229	0	168
	REC	1,204	29,225	758		REC	367	51,884	188
RU	EVAL	341	5,538	321	FA	EVAL	186	21,081	222
	SYS	5,823	14,664	2,834		SYS	7,883	27,186	6,369
	TOTAL	10,804	49,427	4,235		TOTAL	10,665	100,151	6,947
	REDUCED	6,261	8,548	2,215		REDUCED	242	2,453	82

Our final and largest dataset (SYS) consists of a validated subset of the output from both our machine learning-based detection system and earlier versions of the standalone semantic generalization component described which is the focus of this work. Our full dataset consists of vastly more examples that have not been validated. Those that have been validated were selected using an $ad\ hoc$ active learning setting that considered a variety of characteristics – including similarity to existing annotations, target/source diversity, relation diversity, system confidence, and disagreement between the ML system and the semantic generalization components.

The datasets labeled "REDUCED" in Table 2 represent a subset of the combined dataset (all four of the above) which consists of those only instances that can be represented using a pseudo-semantic relation – i.e., metaphors (or literal utterances) with subj-pred, obj-pred, or adj-noun relations. Only this subset can be used and tested crosslingually.

For each dataset, annotators were asked to judge metaphoricity according to criteria comparable to the MIP annotation guidelines [23]. Following the insights of Dunn [6], we have instructed the annotators to employ a four-point metaphoricity scale corresponding to: (0) no metaphoricity, (1)

possible (or weak) metaphoricity, (2) likely (or conventionalized) metaphoricity, or (3) clear metaphoricity. In some cases, multiple annotators provided scores for individual instances, and so we use the average score across all annotators. Using these averages, we categorize each annotated instance as "metaphoric" (score≥1.5), "non-metaphoric" (score<0.5), or "unclear" (0.5<score≤1.5).

Table 3. *Experiments showing the performance of each target/relation representation alongside the combined "cascade" and a lexical baseline. This represents a 10-fold cross-validation over the combined dataset for a given language. For those using the R_{SEM}, the REDUCED dataset it used. The numbers reported above correspond to the F-measure for detecting metaphor. When there is a range specified (low/high), it is due to the way that we are categorizing annotations marked as "unclear". On the low end, annotated examples labeled "unclear" are ignored entirely, while on the high end, any "unclear" examples are labeled as "metaphor" by the system to improve precision. Recall is unaffected by this distinction*.

Target/Relation	ENGLISH	SPANISH	RUSSIAN	FARSI	**Target/Relation**	ENGLISH	SPANISH	RUSSIAN	FARSI
CASCADING	**0.92**	**0.90/0.91**	**0.88**	**0.86/0.87**	BASELINE	0.73	0.76	0.53	0.78/0.79
T_{LEX}, R_{DEP}	0.91	**0.90/0.91**	0.87	0.85/0.86	T_{LEX}, R_{SEM}	0.89/0.90	0.90/0.91	0.87/0.88	0.62
T_{DOM}, R_{DEP}	0.86/0.87	0.87/0.88	0.82/0.83	0.82/0.83	T_{DOM}, R_{SEM}	0.84/0.85	0.87/0.88	0.82	0.60/0.61
T_{SYN}, R_{DEP}	0.87	0.86	0.82	0.80/0.81	T_{SYN}, R_{SEM}	0.86	0.87/0.88	0.81	0.56
T_{SEM}, R_{DEP}	0.83/0.84	0.75/0.76	0.71	0.78/0.79	T_{SEM}, R_{SEM}	0.79	0.67/0.68	0.65/0.66	0.52
T_{ABS}, R_{DEP}	0.82/0.83	0.74/0.75	0.54	0.75/0.77	T_{ABS}, R_{SEM}	0.78/0.79	0.65/0.66	0.47/0.48	0.51/0.52

In deriving our semantic patterns (cf. Section 3), we have not made use of instances labeled "unclear".

5. EXPERIMENTS

In order to highlight the contributions of our approach to semantic generalization for metaphor detection, we have carried out two experiments. First, we evaluate the performance of the semantic patterns derived for individual target/relation representation pairs as well as the performance of the full system in the cascading framework described in Section 3.3. Then, we determine the extent to which annotations from one or more

languages can be applied to the task of metaphor identification in a separate language for which no annotations are available.

5.1. *Monolingual generalization experiments*

In our first experiment, we test the ability of our semantic generalization component (using each of the target/relation representations described in 3.1) to detect metaphors in a monolingual setting. We compare each target/relation representation (used in isolation) against both our cascading combination of patterns described in Section 3.3 and a fully lexical baseline. This baseline consists of the following: for each example in the test fold, we find an exact lexical match of the tuple ($S,\ T_{LEX},\ R_{DEP}$) – meaning the hierarchy was not used – and then apply the most common metaphoricity decision from our annotations. If no lexical matches are found, it is labeled as "unclear".

We have performed our experiment over the combined datasets described in Section 4 using a 10-fold cross-validation – that is, for evaluation against each fold of the data, we develop our semantic patterns over the remaining 9 folds. We report the results of these experiments in Table 3.

The lexical baseline system predictably resulted in very high precision (> 95%) with a comparatively low recall (appx. 50%). For each language, the cascading combination system performed best, highlighting the advantage associated with our overall methodology. Among the individual target/relation pairs, performance was comparable (slightly better) using the raw dependency relation compared to using the pseudosemantic relations. For the target representation, the lexical (T_{LEX}) representation resulted in the best performance followed by the manual domain-level term groupings (T_{DOM}) and the synsets (T_{SYN}). The remaining representations – WordNet semantic categories (T_{SEM}) and the abstract/concrete noun dichotomy (T_{ABS}) – resulted in lower performance due to their coarseness, especially for verbs and adjectives.

Table 4. *Experiments in applying annotations from the other three languages to a given language using cross-lingual target/relation representations*

	ENGLISH	SPANISH	RUSSIAN	FARSI
RECALL	.28	0.48	0.16	0.20
PRECISION	0.72/0.78	0.58/0.72	0.72/0.75	0.26/0.29
F-MEASURE	0.40/0.41	0.53/0.57	0.26	0.22/0.24

5.2. *Cross-lingual generalization experiments*

In our second experiment, we attempt to detect metaphors in the combined dataset of one language using patterns developed from the datasets of the remaining three languages. That is, we make use of no native-language annotations in determining metaphoricity. In particular, these experiments make use of only the "REDUCED" dataset from Section 4 for which we are able to convert each annotation's source/target relation into a cross-lingual, pseudo-semantic relation. All target representations (except for the T_{LEX}) can be applied cross-lingually. The results of these experiments are shown in Table 4.

For English, Spanish, and Russian, the precision of the resulting system at detecting metaphor is above 70%, while recall for these three languages ranges from 16-48%. We believe that this represents good out-of-the-box performance on a novel language with no annotated data. For Farsi, on the other hand, precision is much poorer, but it is difficult to draw conclusions, due to the small size of the "REDUCED" dataset in this language.

6. CONCLUSION

In this work, we have presented a novel approach to the generalization of metaphoricity annotations across a semantic hierarchy. We have clearly shown the advantage of generalizing sources throughout this hierarchy with an F-measure above 0.85 for four languages in 10-fold cross-validation experiments over a very large annotated dataset of metaphoricity. This is compared to a purely lexical baseline with F-measure between 0.53-0.79 across all languages. In a monolingual setting, the benefits

associated with generalizing the targets across the semantic hierarchy are less clear. We theorize that this can mostly be attributed to the restricted number of domains and target lexical items that are represented in our dataset and that this type of generalization would have a more significant effect when applied to novel domains. Likewise, the pseudosemantic relations we propose were not shown to outperform raw dependency relations within our monolingual datasets.

However, we have shown that the pseudo-semantic relations are able to be applied to the task of cross-lingual metaphor detection. For three of our languages, we were able to detect metaphors with over 70% precision and a recall ranging from 16-48%. This clearly shows the utility of our approach in tackling the task of metaphor detection in novel languages with a minimal development cost.

In future work, we intend to apply our approach to semantic generalization to a variety of novel domains to better explore the effect of target-level generalization. In addition, we hope to supplement WordNet with additional pseudo-semantic categories derived using the distributional similarity of terms for use in languages where there is no corresponding version of WordNet or where its quality is poor. Finally, we plan to analyze the performance of our semantic generalization component in an active (or co-active) learning framework in combination with a feature-based metaphor detection system.

ACKNOWLEDGMENTS

This research is supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense US Army Research Laboratory contract number W911NF-12-C-0025. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoD/ARL, or the U.S. Government."

REFERENCES

1. Bogdanova, D. 2010. A framework for figurative language detection based on sense differentiation. In proceedings of the *ACL 2010 Student Research Workshop* (pp. 67-72), Association for Computational Linguistics.
2. Bracewell, D., Tomlinson, M. & Mohler, M. 2013. Determining the conceptual space of metaphoric expressions. In *Computational Linguistics and Intelligent Text Processing* (pp. 487-500), Springer.
3. Bracewell, D., Tomlinson, M., Mohler, M. & Rink, B. 2014 A tiered approach to the recognition of metaphor. In *Computational Linguistics and Intelligent Text Processing*.
4. Broadwell, G. A., Boz, U., Cases, I., Strzalkowski, T., Feldman, L., Taylor, S., Shaikh, S., Liu, T., Cho, K. & Webb, N. 2013. Using imageability and topic chaining to locate metaphors in linguistic corpora. In *Social Computing, Behavioral-Cultural Modeling and Prediction* (pp. 102-110), Springer.
5. Dunn, J. 2013. What metaphor identification systems can tell us about metaphor-in-language. *Meta4NLP 2013* (p. 1).
6. Dunn, J. 2014. Measuring metaphoricity. In proceedings of the *52nd annual meeting of the Association for Computational Linguistics* (pp. 745-751), Vol. 2, Association for Computational Linguistics Stroudsburg, PA.
7. Fass, D. 1991. met*: A method for discriminating metonymy and metaphor by computer. *Computational Linguistics*, 17/1, 49-90.
8. Fellbaum, C. 1998. *WordNet, An Electronic Lexical Database*, The MIT Press.
9. Hovy, D., Srivastava, S., Jauhar, S. K., Sachan, M., Goyal, K., Li, H., Sanders, W. & Hovy, E. 2013. Identifying metaphorical word use with tree kernels. *Meta4NLP 2013* (p. 52).
10. Krishnakumaran, S. & Zhu, X. 2007. Hunting elusive metaphors using lexical resources. In proceedings of the *Workshop on Computational approaches to Figurative Language* (pp. 13-20), Association for Computational Linguistics.
11. Lakoff, G. 1993. The contemporary theory of metaphor. *Metaphor and Thought*, 2, 202-251.
12. Lakoff, G. 1994. *Master Metaphor List*. University of California.
13. Lakoff, G. & Johnson, M. 1980. *Metaphors we live by*, 111. Chicago London.
14. Levin, L., Mitamura, T., Fromm, D., MacWhinney, B., Carbonell, J., Feely, W., Frederking, R., Gershman, A. & Ramirez, C. 2014.

Resources for the detection of conventionalized metaphors in four languages. In proceedings of the *Ninth Language Resources and Evaluation Conference (LREC)*, Reykjavik, Iceland.

15. Li, H., Zhu, K.Q. & Wang, H. 2013. Data-driven metaphor recognition and explanation. *TACL 1* (pp. 379-390).
16. Li, L. & Sporleder, C. 2010. Linguistic cues for distinguishing literal and non-literal usages. In proceedings of the *23rd International Conference on Computational Linguistics*: Posters (pp. 683-691), Association for Computational Linguistics.
17. Martin, J. H. 1994. MetaBank: A knowledge-base of metaphoric language conventions. *Computational Intelligence*, 10/2, 134-149.
18. Martin, J. 1990. A computational model of metaphor interpretation. Academic Press Professional, Inc.
19. Mason, Z. 2004. CorMet: A computational, corpus-based conventional metaphor extraction system. *Computational Linguistics*, 30/1, 23-44.
20. Mohler, M., Bracewell, D., Hinote, D. & Tomlinson, M. 2013. Semantic signatures for example based linguistic metaphor detection. *Meta4NLP 2013* (p. 27).
21. Nayak, S. & Mukerjee, A. 2012. A grounded cognitive model for metaphor acquisition. In *AAAI*.
22. Neuman, Y., Assaf, D., Cohen, Y., Last, M., Argamon, S., Howard, N. & Frieder, O. 2013. Metaphor identification in large texts corpora. *PloS one*, 8/4, e62343.
23. Pragglejaz Group. 2007. MIP: A method for identifying metaphorically used words in discourse. *Metaphor and symbol*, 22/1, 1-39.
24. Sardinha, T. B. 2010. A program for finding metaphor candidates in corpora. *ESPecialist*, 31/1, 49-67.
25. Schulder, M. & Hovy, E. 2014. Metaphor detection through term relevance. *ACL 2014* (p. 18).
26. Sporleder, C. & Li, L. 2009. Unsupervised recognition of literal and non-literal use of idiomatic expressions. In proceedings of the *12th Conference of the European Chapter of the Association for Computational Linguistics* (pp. 754-762), Association for Computational Linguistics.
27. Tsvetkov, Y., Boytsov, L., Gershman, A., Nyberg, E. & Dyer, C. 2014. Metaphor detection with cross-lingual model transfer. In: *Proceedings of ACL*.
28. Turney, P. D., Neuman, Y., Assaf, D. & Cohen, Y. 2011. Literal and metaphorical sense identification through concrete and abstract

context. In proceedings of the *2011 Conference on the Empirical Methods in Natural Language Processing* (pp. 680-690).
29. Wilks, Y., Galescu, L., Allen, J. & Dalton, A. 2013. Automatic metaphor detection using large-scale lexical resources and conventional metaphor extraction. *Meta4NLP 2013* (p. 36)

MICHAEL MOHLER
LANGUAGE COMPUTER CORPORATION,
RICHARDSON, TX.
E-MAIL: <MICHAEL@LANGUAGECOMPUTER.COM>
HTTP://WWW.LANGUAGECOMPUTER.COM

MARC TOMLINSON
LANGUAGE COMPUTER CORPORATION,
RICHARDSON, TX.
E-MAIL: <MARC@LANGUAGECOMPUTER.COM>
HTTP://WWW.LANGUAGECOMPUTER.COM

BRYAN RINK
LANGUAGE COMPUTER CORPORATION,
RICHARDSON, TX.
E-MAIL: < BRYANG@LANGUAGECOMPUTER.COM>
HTTP://WWW.LANGUAGECOMPUTER.COM

International Journal of Computational Linguistics and Applications, Vol. 6, No. 2, 2015, pp. 141–158
Received 25/01/2015, Accepted 27/02/2015, Final 20/12/2015.
ISSN 0976-0962, http://ijcla.bahripublications.com

A #hashtagtokenizer for Social Media Messages

VLÁDIA PINHEIRO
RAFAEL PONTES
VASCO FURTADO
Universidade de Fortaleza, Ceará, Brasil

ABSTRACT

In social media, mainly due to length constraints, users write succinct messages and use hashtags *to refer to entities, events, sentiments or ideas.* Hashtags *carry a lot of content that can help in many tasks and applications involving text processing such as sentiment analysis, named entity recognition and information extraction. However, identifying the individual words of a* hashtag *is not trivial because the traditional POS taggers typically consider it as a single token, despite the fact that it might contain multiple words, e.g.* #fergusondecision*,* #imcharliehebdo*. In this work, we propose a generic model for* hashtagtokenisation *that aims to split up one* hashtag *into several tokens corresponding to each individual word contained in it (e.g. "*#imcharliehebdo*" would become four tokens, "#", "*i*", "*am*" and "*Charlie Hebdo*"). Our* hashtagtokenizer *is based on a machine learning segmentation method for Chinese language and makes also use of Wikipedia as encyclopedic knowledge base. We have evaluated the inference power of our approach by comparing the tokens produced by our approach to those produced by human taggers. The results demonstrated the good accuracy and applicability of the proposed model for general-purpose applications.*

Keywords: Social media, information extraction, tokenization, hashtag

1. INTRODUCTION

Social medias are extremely popular and thus generate so much user-generated data. These data are interesting from a computational linguistic point of view to investigate the potential of extracting useful information from this data. In the field of natural language processing (NLP), a large number of tools rely on the availability of morphosyntactic or part-of-speech (POS) information about texts of microblogs and of social networks reviews [1]. In these social media, users post texts ina very specifically way that requires special handling. Mainly due to the restriction on the length of the messages, users write succinct messages and use *hashtags* to refer to another entities or events in order to facilitate the text retrieval and to express sentiments and ideas. The use of *hashtags*[1] is a popular way to give the context of a tweet or the core idea expressed in the tweet [2].For example, the *hashtag* #savethenhs reads as 'savethenational health service.'[2]

Hashtags carry a lot of content that can help in many tasks and applications involving NLP such as sentiment analysis [3,4], named entity recognition, information extraction and retrieval [5], prediction of the spread of ideas for marketing purposes [2,6]. Maynard and Greenwood [3] claim that much useful sentiment information is contained within *hashtags* and that we can make use of the information contained within them for sentiment detection. For example, we can recognize positive and negative words within a *hashtag*.

However, identifying the individual words of a *hashtag* is not trivial because the traditional POS taggers typically tokenize the *hashtags* as a single token, although they contain multiple words, e.g. #notreally, #fergusondecision, #imcharliehebdo. General purpose tokenizers need to be adapted to work correctly on social media, in order to handle specific tokens like URLs, *hashtags*,

---

[1] A *hashtag* is a sequence of non-whitespace characters preceded by the hash character "#". For example, #healthcarereform is a hashtag.

[2] Note that National Health Service (NHS) stands for the four publicly funded health care systems in the countries of the United Kingdom.

user mentions in microblogs, special abbreviations, and emoticons. Few works focus on decomposing the *hashtag* into tokens. Systems those somehow aim to discover the individual tokens of a *hashtag* or are incipient or intend to specific situations [2,3]. For example, Maynard and Greenwood [3] use gazetteers and a dictionary of common slang for detecting sarcasm within hashtags.

In this work, we propose a model for *hashtag*tokenisation that aims to split up a *hashtag* (commonly defined by traditional POS taggers) in several tokens corresponding to each individual word contained in the hashtag (e.g. "#imcharliehebdo" becomes four tokens, "#", "i", "am" and "Charlie Hebdo"). We argue that this model can be coupled in a traditional POS tagger to leverage the potential of *hashtag* analysis in NLP processors. Our #hashtagtokenizer is based on an unsupervised word segmentation approach, used for Chinese language [7], plus a linguistic and worldwideknowledge approach which uses a lexicon and anencyclopedic knowledge base. Specifically, we propose a segmentation algorithm which generates the possible segmentation options $s_j = w_1 \oplus \ldots \oplus w_n$ of a hashtag***h***, where w_i is a valid word hypothesis. In order to do so, we have searched in a lexicon or vocabulary of the language used and in Wikipedia. This strategy addresses the observation that several *hashtags* are composed of people's names, places, brands, etc, and may be directly solved through world knowledge bases (e.g. #iphone, #androidgames). To solve the best segmentation options for a *hashtag**h***, our approach utilizes a word induction score inspired on the work of [7]. This score is calculated from a value of external boundaries, which captures the limits to the left and right of valid word hypothesis through a co-occurrence matrix. It also makes use of a value of internal boundary, which indicates the most likely separation point of a determined word combination.

We use the implemented #hashtagtokenizer in a list of the most frequent *hashtags* from a corpus collected from Twitter in December, 2014. We conducted experiments not only to analyze the accuracy of the proposed model but also the challenges to discover the components words of a *hashtag*. The results

demonstrate the good accuracy and applicability of the proposed model for general-purpose applications.

2. RELATED WORK

Gimpel et al. [8] address the problem of POS tagging for English data from Twitter. Starting from scratch, they developed an English Twitter-specific tag set. Using this tag set, they manually corrected English tweets that were annotated using Stanford POS tagger [9] and additionally developed features to build a machine learning classifier that tags unseen tweets. Avontuur et al. [1]propose a similar approach for Dutch tweets. TwitIE [5] is an open-source NLP pipeline customized to microblog text at every stage that comprises: language identification, tokenizer, normalizer, POS tagger and Named Entity Recognition. In this work, the authors recognize that general purpose tokenizers need to be adapted to work correctly on social media, in order to handle specific tokens like URLs, hashtags (e.g. #nlproc), user mentions in microblogs (e.g. @GateAcUk), special abbreviations (e.g. RT, ROFL), and emoticons. TwitIEtokenizer follows Ritter's tokenisation scheme [10] and treats *hashtags* and user mentions as two tokens (i.e., '#' and 'nlproc' in the above example) with a separate annotation HashTag covering both. All the aforementioned works focus on identifying one specific tag (e.g. /HASH) instead of a complete decomposition of a hashtag into several tokens.

In [2] the analysis of a hashtag content has been done for understanding meme propagation. The authors prepared a regression model using features aboutthe length of the hashtag(characters and words) and the lexical items. These lexical items are manually segmented. After that therole of each one of them is discovered through searches in datasets of names, celebrities, countries, holidays, etc. It is worth mentioning that the focus of this work is not automatically identifying the lexical items of a *hashtag*.

Maynard and Greenwood [3] have compiled a number of rules, which enable to improve the accuracy of sentiment analysis when sarcasm is known to be present. In particular, they

considered the effect of sentiment and sarcasm contained in hashtags, and they have developed a hashtagtokeniser for GATE [11], so that sentiment and sarcasm found within hashtags can be detected more easily. First, they try to form a token match against GATE's gazetteers (vocabulary, locations, organizations etc.), and against an edited dictionary of common slang words from [5]. According to their experiments, the hashtagtokenization achieves 98% precision. Unfortunately this work cannot be used as a parameter of comparison because the data used is not available and the impact of the gazeteers and dictionaries could not be evaluated.

3. THE #HASHTAGTOKENIZER MODEL

In this work, we propose a model for hashtagtokenisation that aims to split up the single token of a *hashtag* in several tokens corresponding to each individual word contained in it (e.g. "#imcharliehebdo" becomes four tokens, "#", "i", "am" and "Charlie Hebdo"). We argue that this model can be coupled in a traditional POS tagger to leverage the potential of *hashtag* analysis in NLP processors. Our #hashtagtokenizer is based on two approaches: (1) a lexicon and a world knowledge approach to identify tokens that express valid words of a lexicon of a language, or names of people, organizations, brands, locations, etc., in a encyclopedic knowledge base such as Wikipedia; (2) an unsupervised word segmentation approach of proposed in [7], used for languages where there is no visual representation of word boundaries in a text (e.g. Chinese or Japanese languages).

3.1. *Model overview*

Figure 1 presents the generic pipeline of our proposed model for hashtag tokenization.

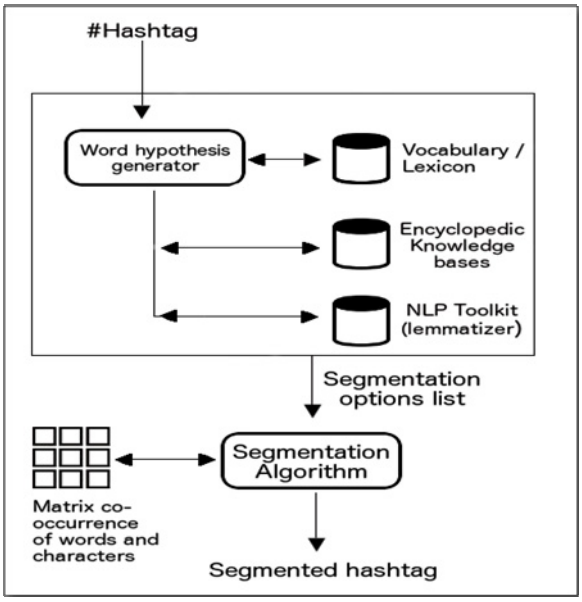


Figure 1. *The HashtagTokenizer Pipeline*

The **Word Hypothesis Generator** receives a *hashtag**h*** (typically a single token) and tries to decompose it in a sequence of valid word hypothesis. A word hypothesis is valid whether it exists in a vocabulary or knowledge base. The set of w_i valid words hypothesis are then sent to the ***Segmentation Algorithm*** that searches for the sequence that maximizes the *WordRank* score. Such a sequence is defined as the segmentation of ***h***. We are going to detail this process as following.

3.2. *Segmentation algorithm*

We developed a Viterbi-like algorithm in order to the search for the optimal segmentation of a hashtag***h*** (an utterance of continuous characters). Firstly, the segmentation algorithm is used to find the possible segmentationsj = $w_1 \oplus \ldots \oplus w_n$ of***h***, where w_i is a valid word hypothesis. Thereafter, it searches for the segmentation option that maximizes the following objective function (Formula 1):

$$s(h) = \text{argmax}_{w_i + \ldots + w_n} WR(w_i) \quad (1)$$

where $WR(w)$ is the WordRank score of a valid word w. In other words, the resulting segmentation of hashtag***h*** is the segmentation option $w_1 \oplus \ldots \oplus w_n$ with the highest function value.

3.2.1. *Generating the segmentation options*
Given an unsegmented *hashtag****h***, we may retrieve "naive" word hypotheses by considering all the characters sequences to be word hypotheses. For example, for the hashtag "#fergusondecision", we may retrieve the following naive segmentation options - "#", "fer", "gus", "ond", "eci", "sion" - among many other possible combinations of character sequences. In order to reduce the number of segmentation options of ***h***, we use the following strategies:

1. Search in encyclopedic knowledge bases for the word hypothesis w, formed by the complete *hashtag* without the character hash "#". In case of wbeing founded, it is generated as the resulting segmentation of ***h***. This strategy is a response to the observation that *hashtags* are formed by names of people, entities, places, brands, etc, which can be directly solved by world knowledge bases (e.g. #youtube, #iphone, #android). For instance, for the *hashtag* '#android', the segmentation option s_1 = 'android' is generated and searched for in a database such as Wikipedia. Since this verbatim is found on Wikipedia, the segmentation of a***h*** is simply w_1 = '#' $\oplus$ w_2 = 'android'.
2. In an interactive algorithm, ***h***is divided in all possible sequences of word hypothesis w_i, and for every w_i, the process checks if:
   a. It is a valid word from a dictionary or a language lexicon;
   b. Its lemma is a valid word from a dictionary or a language lexicon

c. It is a verbatim from an encyclopedic knowledge base, e.g Wikipedia.

In case of any of the situations aforementioned being true to all word hypothesis w_i, they are considered *valid word hypothesis* and the corresponding combination $w_i \oplus \ldots \oplus w_n$ is generated as a possible option for segmentation of ***h***. For instance, for the hashtag***h*** = 'good', the possible sequences for word hypothesis are the following:

s_1 = *'g' ⊕ 'o' ⊕ 'o' ⊕ 'd'*
s_2 = *'g' ⊕ 'o' ⊕ 'od'*
s_3 = *'g' ⊕ 'oo' ⊕ 'd'*
s_4 = *'g' ⊕ 'ood'*
s_5 = *'go' ⊕ 'o' ⊕ 'd'*
s_6 = *'go' ⊕ 'od'*
s_7 = *'goo' ⊕ 'd'*
s_8 = *'good'*

In this example the segmentation option s_4 is rejected because all word hypothesis 'g' e '*ood*' have not matchany of the mentioned conditions (a), (b) or (c). Options 5 was also rejected because the word hypothesis '*od*' failed, although '*go*' is a valid word in English. Options 8 was accepted since its word hypothesis '*good*' is a valid English word.

3. Finally, a list of segmentation options in the form of $w_i \oplus \ldots \oplus w_n$ is generated. It is worth noting that the use of a reliable encyclopedic knowledge and with a currently updated database allows for a more robust and in-depthmodel. Furthermore, it enables for some word hypothesis to have their meaning clarified since they have been associated to a Wikipedia concept, thus facilitating the Word Sense Disambiguation process.

3.2.2. *Computing the WordRank (WR) score*

We propose a word induction criteria based on [7], which propose the WordRank. The intuition of their idea is that word

boundaries between adjacent words indicate the correctness of each other, i.e., if a word hypothesis has a correct (or wrong) word boundary, we may infer that its neighbor would simultaneously have correct (or wrong) word boundary at its corresponding side. This idea is similar to the ideas of Firth [12] that "You shall know a word by its company" and the distributional hypothesis of Harris [13], that words will occur in similar contexts only if they have similar meanings.

All this motivated us to construct a matrix of co-occurrence of words which expresses how often a word co-occurs in a corpora to other words from a window of [-n,+n] words. Latent Semantic Models are also based on information about the context of use [14]. Among those, Hyperspace Analog to Language (HAL) [15] is a model that acquires representations of meaning by capitalizing on large-scale co-occurrence information inherent in the input language stream. The basis for the methodology to represent the meaning of HAL is to develop a matrix of word co-occurrence values for a given vocabulary, from a large text corpus, using a window size. The smallest useable window would be [-1,+1] words, corresponding to only the immediately adjacent words. By constructing this matrix it was able to express the frequency of occurrence of words adjacent to the left and right of a given word w. Table 1 presents an example of a matrix of co-occurrence of words for an input corpus "*City Employees Plan Walkout for Police Reform. City employees are mobilized*", with window [-1,+1], which values express the frequency of a given word w_k to the left (and right) of a word w_l. By using the matrix row as guidance, the word 'city' co-occurs 02 (twice) to the left of the word '*employee*'. By using the matrix column as guidance, the word '*employee*' co-occurs 02 (twice) to the right of the word 'city'.

The matrix of co-occurrence of words is similar to the link structures proposed in [7], considering that it also expresses the connections between words adjacent to the left and right of a given word w.

Table 2. *Matrix of co-occurrence of words for an input corpus "City Employees Plan Walkout for Police Reform. City employees are mobilized", with window [-1,+1]*

	city	employee	plan	walkout	for	police	reform	are	mobilize
city		2							
employee			1					1	
plan				1					
walkout					1				
for						1			
police							1		
reform									
are									1
mobilize									

Having constructed the matrix of co-occurrence of words, it was possible to calculate the Left-side Information (LI) and Right-side Information (RI) for a valid word hypothesis w, according to Formula (2) and (3), respectively.

$$LI(w) = \sum_{l \in s_j} RI(l) \tag{2}$$

$$RI(w) = \sum_{r \in s_j} LI(r) \tag{3}$$

where,

- ***RI'(l)***is the frequency of co-occurrence of all words ***l*** in s_j which are present to the left of w (or in which w co-occurs to the right of l). In the co-occurrence matrix, it is the value of the cell M_{ij}, where ***i*** is the line of word ***l*** and ***j*** is the column of the word ***w***.
- ***LI'(r)***is the frequency of co-occurrence of all words ***r*** in s_j which are present to the right of w (or in which w co-occurs to the left of l). In the co-occurrence matrix, it is the value of the cell M_{ij}, where ***i*** is the line of word ***w*** and ***j*** is the column of the word ***r***.

Finally, an External Value (EV) for a valid word hypothesis w is calculated by Formula (4).

$$EV(w) = LI(w) * RI(w) \quad (4)$$

According to [7], it is not enough to represent the goodness of a word hypothesis using only information of external boundaries, which, in this paper, is based on the co-occurrence frequency with words adjacent to the left and right (RI and LI). The justification is that word combinations are prioritized, since they might have a high EV value as long as the internal limits of the word hypothesis are ignored. Considerthe following example where the word hypothesis w = '*thatdog*', which external limits to the right and left were well defined. However, it is formed by two words 'that' and 'dog'. Thus we need to find the internal boundary between 't' e 'd' considering that the set of letters 'td' occurs more rarely in an English word.

To solve this problem, the Internal Value (IV) for a valid word hypothesis w is calculated by Formula (5), based on Mutual Information (MI) that measures the combining degree of pairs of adjacent characters [16]. It has been reported that a high MI value indicates a good chance of two characters combining together, whereasa low MI value indicates a word internal boundary between the two characters.

$$IV(w) = \min_{i=1|L-1}(MI(c_i, c_{i+1})) \quad (5)$$

Where L is the length of a given word hypothesis w and c_i is the i_{th} character of w. The Internal Value (IV) assumes the lowest MI value among all character pairs of the word hypothesis was long as it is the most likely point of the internal boundary of w.

Finally, we calculate the WordRank (WR) score for a word hypothesis w, according to [7, p.869], as follows:

$$WR(w) = EV(w) * f(IV(w)) \quad (6)$$

where,

- *f(x)* is the auxiliary function for optimal performance. Chen, Xu and Chang [7] use the functions polynomial $(f(x)=x^{\alpha})$ and exponential $(f(x)=\beta^{x})$ with parameters α and β. For example, for English, some experiments indicate that $\alpha = 4.4$ and $\beta = 4.6$ are optimal values.

4. EXPERIMENTAL EVALUATIONS

We have chosen the social media Twitter as a source of *hashtags* to evaluate our model. Twitter is a popular microblogging platform and users post *hashtags* to give the context of a tweet, mainly due to the restricted size of messages of only 140 characters. A study of 1.1 million tweets established that 26% of English tweets have a URL, 16.6% – a *hashtag*, and 54.8% – a user name mention [17].

The goals of our evaluationwere to analyze the accuracy of the proposed model and identify which are the main challenges in accomplishing this task. Our hypotheses for investigation are: (1) that the use of encyclopedic knowledge improves the accuracy of the approach; (2) the use of a lexicon and language vocabulary reduces the number of segmentation optionsand optimizes the model performance.

4.1. *DataSet and gold standard*

Using the Twitter 4J API[3], we collected 93000 Twitter posts with *hashtags* sent during one week in December, 2014. Overall it is 122705 *hashtags*. Table 2 presents the hashtags frequency distribution. Note that402distinct hashtags were used more than 100 times in the corpus and 120264 were mentioned from 1 to 19 times. We have than selected as the core of our dataset the 402more frequent hashtags.

[3] http://twitter4j.org/en/index.html

Table 3. *Frequency distribution of* hashtags *in the Twitter corpus.*

Frequency Range	Number of hashtags
>=100	402
80-99	113
60-79	235
40-59	398
20-39	1293
1-19	120264

To be able to evaluate the output of the *hashtag*tokenizer, a Gold Collection of correctly segmented *hashtags* is required. This allows for a comparison of the output of the implemented #hashtagtokenizer against this Gold standard. As we are not aware of any Gold Standard for this task, we had to manually build it. Two computer science graduate students, English-speakers, affinity with Twitter and adopters of *hashtags*, created the Gold standard. They agreed in 97% of the tokens. A third student resolved the cases of disagreements.

4.2. *Experimental setup and results*

We developed #hashtagtokenizerembedding the segmentation algorithm and the computation of the score (see Section 3)as well as the following resources:

- English vocabulary used in [18] with 27000 valid words
- Lemmatizer - The Stanford CoreNLP Toolkit [19]
- Encyclopedic knowledge base – DBPedia 3.9
- English corpus for the matrix of co-occurrence – Stanford WebBase Project [20]

Three evaluation scenarios were created, which are presented below.

- **BASELINE** approach – hashtag segmentation based on capitalization of words. For instance, #FergusonDecision, generates two tokens based on the capital letters ‘F’e ‘D’ which are the indicatives of boundaries to left and to right of each token;

- **VOCabulary** + **WR** – segmentation uses a vocabulary of valid English words [18], the lemmatizer of the Stanford CoreNLP toolkit [19], and computes the WordRank Score (without Wikipedia) .
- **VOCabulary** + **DBPedia** + **WR** – the same as the previous using also Wikipedia.

Table 3 presents the results in each scenarioin terms of accuracy (based on the number of hashtags that have been correctly tokenized when compared to the Gold Standard).

Table 4. *Accuracy of the #hashtagtokenizerin the three evaluation scenarios*

Evaluation Scenario	Accuracy
BASELINE	63.9%
VOC+WR	58.3%
VOC+DBP+WR	73.2%

When we analyze the results, we can see that scenario VOC+DBP+WR (Vocabulary+DBPedia+WordRank) shows a gain of 25.5% of accuracy compared to scenario VOC+WR (Vocabulary+WordRank – without DBPedia). This result fortifies our claim that the use of encyclopedic knowledge improves the accuracy of hashtags tokenization (our first work hypothesis). Indeed the use of Wikipedia as a knowledge base is more and more a consensus since its dynamism, large scope and reliability [21]. Our analyses have shown that the mention to people, entities, brands, places and events is frequent in *hashtags*. Moreover, these emerge as memes requiring knowledge bases with high capacity to treat with volatile terms. Another benefit of using Wikipedia is that the meaning of the tokens is already defined leveraging the word sense disambiguation task. We know that the BASELINE scenario (using capitalization as a boundary for word separation) is naïve and useful only as an initial reference. However we did not find available similar approaches for comparisons.

Another formulated hypothesis is that the use of a lexicon and of a vocabulary of the language narrow the number of

segmentation options thus optimizing the model. In fact, when we remove the vocabulary of the process of tokenization, the number of word hypothesis increases substantially. Without this resource the method has generates 16,486,141 options instead of 284,506 for the case the vocabulary is used. For the some hashtags such as #sledgehammervideopremiere, #mentionpeopleyouarethankfulfor and #asklittlemixalittlequestion more than 3 million options were generated.

We have also done a qualitative analysis from the cases in which the method failed. Most of the problems occur with hashtags containing numbers such as #63notout, #16days, #iphone6, #500aday (almost 20% of the errors). The other major source of bad inferences are due to acronyms such as #superstarRK, #AFCvBOR, #SEAvsSF) (17% of the errors). These problems will guide our future investigations.

5. CONCLUSION

In this paper we propose a generic model for *hashtag* tokenization based on a lexicon and a world knowledge bases, and on an unsupervised word segmentation algorithmused for languages where there is no visual representation of word boundaries in a text (e.g. Chinese or Japanese languages). The *Hashtag* Tokenizerpipeline searches, firstly, to identify tokens that express valid words of a lexicon, or names of people, organizations, brands, locations, etc., in an encyclopedic knowledge base such as Wikipedia. Thereafter, it searches for the segmentation option that maximizes the WordRank score, which captures the limits to the left and right of valid word hypothesis through a co-occurrence matrix of co-occurrence of words (external boundaries), combined with a value of internal boundary, which indicates the most likely separation point of a determined word combination.

Our research hypotheses were that the use of encyclopedic knowledge improves the accuracy of the approach, and that the use of a lexicon and language vocabulary optimizes the model performance. We have evaluated the accuracy of the proposed

model by comparing the tokens produced by our approach to a Gold Standard produced by human taggers. The best evaluation scenario, whichused an English vocabulary, the DBpedia as encyclopedic knowledge base, and the WordRank score, presented an accuracy of 73.2%, with a gain of 25.5% of accuracy compared to scenario without DBPedia. This experimental evaluation provided real scenarios of assessing the challenges to discover the components words of a *hashtag*. Almost 37% of bad inferences were due *hashtags* containing numbers and acronyms. These problems will guide our future investigations.

REFERENCES

1. Avontuur, T., Balemans, I., Elshof, L., van Noord, N. & van Zaanen, M. 2012. Developing a part-of-speech tagger for Dutch tweets. *Computational Linguistics in the Netherlands Journal*, 2, 34-51.
2. Tsur, O. & Rappoport, A. 2012. What's in a hashtag?: Content based prediction of the spread of ideas in microblogging communities. In proceedings of the *WSDM'12, The Fifth ACM International Conference on Web Search and Data Mining* (pp. 643-652).
3. Maynard, D. & Greenwood, M. A. 2014. Who cares about sarcastic tweets? Investigating the impact of sarcasm on sentiment analysis. Diana Maynard. *Proceedings of LREC 2014*, Reykjavik, Iceland.
4. Pak, A. & Paroubek, P. 2010. Twitter as a corpus for sentiment analysis and opinion mining. *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC 2010)* (pp. 1320-1326), Valletta, Malta.
5. Bontcheva, K., Derczynski, L., Funk, A., Greenwood, M.A., Maynard, D., Aswani & N. TwitIE. 2013. An open-source information extraction pipeline for Microblog text. In proceedings of the *International Conference on Recent Advances in Natural Language Processing, ACL*.
6. Cunha, E., Magno, G., Comarela, G., Almeida, V., Gonçalves, M.A., Benevenuto, F. 2011. Analyzing the dynamic evolution of hashtags on Twitter: A language-based approach. *Proceedings of the Workshop on Languages in Social Media* (pp. 58-65), Jun 23-23, Portland, Oregon.

7. Chen, S., Xu, Y. & Chang, H. 2011. A simple and effective unsupervised word segmentation approach. *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, *AAAI 2011*, San Francisco, California, USA, Aug 7-11.
8. Gimpel, K., Schneider, N., O'Connor, B. et al. 2011. Part-of-speech tagging for twitter: Annotation, features and experiments. In proceedings of the *49th Annual Meeting of the Association for Computational Linguistics*: *Short Papers*; Portland, OR, USA, ACL New Brunswick, NJ, USA.
9. Toutanova, K., Klein, D., Manning, C. & Singer, Y. 2003. Feature-rich part- of-speech tagging with a cyclic dependency network. *Proceedings of the HLT-NAACL; Edmonton, Canada, North American Chapter of the Association for Computational Linguistics (NAACL)*, (pp. 252-259).
10. Ritter, A., Clark, S., Mausam & Etzioni, O. 2011. Named entity recognition in tweets: An experimental study. In *proc. Of Empirical Methods for Natural Language Processing (EMNLP)*, Edinburgh, UK.
11. Cunningham, H. et al. 2011. Text processing with GATE (Version 6). University of Sheffield Department of Computer Science. ISBN 0956599311.
12. Firth, J. R. 1968. A synopsis of linguistic theory, 1930-1955. In John R. Firth (Ed.), *Selected Papers of JR Firth*, 1952-59. Indiana University Press, Bloomington (pp. 168-205).
13. Harris, Z. 1968. *Mathematical Structures of Language*. New York, USA: Wiley.
14. Landauer, T. K., Dumais, S. T. 1997. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, 104/2, 211-240.
15. Burgess, C., Livesay, K. & Lund, K. 1998. Explorations in context space: Words, sentences, discourse. *Discourse Processes*, 25, 211-257.
16. Sun, M., Shen, D. & Tsou, B. K. 1998. Chinese word segmentation without using lexicon and handcrafted training data. In proceedings of the *17th International Conference on Computational Linguistics*, 2, 1265-1271.
17. Carter, S., Weerkamp, W. & Tsagkias, E. 2013. Microblog language identification: Overcoming the limitations of short, unedited and idiomatic text. *Language Resources and Evaluation Journal*.
18. Han, L., Kashyap, A. L., Finin, T., Mayfield, J. & Weese, J. 2013. UMBC_EBIQUITY-CORE: Semantic textual similarity systems.

In proceedings of the *Second Joint Conference on Lexical and Computational Semantics. Association for Computational Linguistics*.
19. Manning, C. D., Surdeanu, M., Bauer, J. et al. 2014. The stanford core NLP natural language processing toolkit. In proceedings of *52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.
20. Stanford WebBase project. http://bit.ly/WebBase. Accessed in January, 31, 2015.
21. Zang L. J., Cao, C. & Cao, Y. N. et al. 2013. A survey of commonsense knowledge acquisition. *Journal of Computer Science and Technology*, 28/4, 689-719. DOI 10.1007/s11390-013-1369-6.

VLÁDIA PINHEIRO
PROGRAMA DE PÓS-GRADUAÇÃOEM INFORMÁTICA APLICADA –
UNIVERSIDADE DE FORTALEZA
AV. WASHINGTON SOARES, 1321,
FORTALEZA, CEARÁ, BRASIL.
E-MAIL: <VLADIACELIA@UNIFOR.BR>

RAFAEL PONTES
PROGRAMA DE PÓS-GRADUAÇÃOEM INFORMÁTICA APLICADA –
UNIVERSIDADE DE FORTALEZA
AV. WASHINGTON SOARES, 1321,
FORTALEZA, CEARÁ, BRASIL.
E-MAIL: <RAFAELLPONTES@GMAIL.COM>

VASCO FURTADO
PROGRAMA DE PÓS-GRADUAÇÃOEM INFORMÁTICA APLICADA –
UNIVERSIDADE DE FORTALEZA
AV. WASHINGTON SOARES, 1321,
FORTALEZA, CEARÁ, BRASIL.
E-MAIL: <VASCO@UNIFOR.BR>

International Journal of Computational Linguistics and Applications, Vol. 6, No. 2, 2015, pp. 159–175
Received 16/01/2015, Accepted 27/02/2015, Final 09/09/2015.
ISSN 0976-0962, http://ijcla.bahripublications.com

Combining Textual and Visual Features to Identify Anomalous User-generated Content

LUCIA NOCE
IGNAZIO GALLO
ALESSANDRO ZAMBERLETTI
University of Insubria, Italy

ABSTRACT

Anomaly detection has extensive use in a wide variety of applications, such techniques aim to and patterns in data that do not conform to expected behavior. In this work we apply anomaly detection to the task of discovering anomalies from user-generated content of commercial product descriptions. While most of the other works in literature rely exclusively on textual features, we combine those textual descriptors with visual information extracted from the media resources associated with each product description. Given a large corpus of documents, the proposed system infers the key features describing the behavioral traits of expert users, and automatically reports whenever a newly generated description contains suspicious or low quality textual/visual elements. We prove that the joint use of textual and visual features helps in obtaining a robust detection model that can be employed in an enterprise environment to automatically mark suspicious descriptions for further manual inspection.

Keywords: User-generated commercial content, anomaly detection, visual features, textual features, one-class support vector machine

1. INTRODUCTION

Anomaly detection is an important problem that finds interest and usage in many research areas and application domains, e.g. activity

monitoring, fault diagnosis, satellite image analysis, time-series monitoring, pharmaceutical research, medical condition monitoring, detecting novelties in images, detecting unexpected entries in databases or detecting novelty in text [1-3] to name a few.

The main purpose of anomaly detection systems is to identify and, if necessary, remove anomalous observations from data. An anomalous observation, commonly referred to as *outlier*, is defined as a pattern in data that do not conform to a well-defined notion of normal behavior [2].

According to different application domains, outliers often correspond to important and prosecutable information, and arise because of human error, fraudulent behavior, instrument error, natural deviations in populations or faults in systems, *e.g.* an anomalous credit card transaction could detect fraudulent applications for credit cards; anomalous traffic data could correspond to unauthorized access in computer networks; anomalies in magnetic resonance images may suggest presence of malignant tumors [4-6].

In this manuscript we apply anomaly detection to a novel task, casting the problem of discovering fake or suspicious user-generated descriptions of commercial products as an anomaly detection problem; where an outlier description is one that contains textual or visual elements that differ from a notion of canonical behavior inferred from a large corpus of genuine and high quality handcrafted descriptions.

The task of automatically identifying fake or low quality user-generated content is particularly interesting, as nowadays many websites allow users to post their own content (comments, posts, tweets, digital images, video, audio files, *etc.*) to provide a complete and social user experience; while this may increase the credibility of the website and thus increase the user base, it also exposes the platform to a wide number of possible threats, e.g. fake, low quality or malicious user-generated content may damage the credibility of the website and, in some cases, lead to legal sanctions against the website itself.[1]

[1] TripAdvisor fined $600,000 for fake reviews. http://www.cnbc.com/id/102292002

In this work, we inspect whether it is possible to define a system that automatically and effectively marks potentially fake user-generated content coming from a platform that allows users to buy/sell commercial products and to directly provide product descriptions that may include both textual and media information (images and videos).

Unlike other works in literature that focus exclusively on analyzing the textual information [7-11], in our work we also exploit visual attributes extracted from the images attached to the user-generated content, building an innovative and more complete set of features that better describes the typical characteristics that a canonical high quality product description should possess. Such set of textual and visual features is used to train a machine learning model [12], achieving excellent detection rates and fast recognition times.

2. RELATED WORKS

The topic of this work is strongly related to the following research areas: Anomaly Detection, Image Analysis and One Class Classification.

In literature, anomaly detection is primarily applied on text data as *novelty detection*, with the main aim of detecting novel topics, news stories or events in a collection of documents. Anomalies, or novelties, arise because of a newsworthy event or the presence of a different topic in a particular document, *e.g.* Baker *et al.* [11] address this kind of problem using probabilistic generative models; the more recent work of Blanchard *et al.* [10] approaches novelty detection using semi-supervised models; Mahapatra *et al.* [9] exploit contextual text information to detect anomalies in text data.

In their works, Guthrie *et al.* [7, 8] specialize on finding outliers in documents considering several aspects such as topic, author, genre or emotional tone, and use a combination of stylistic and lexical features to detect anomalies. Their unsupervised model can reliably identify outliers composed of 1000 or more words [8]; a paragraph of text is classified as

anomalous when it is irregular, or when it substantially deviates from its surroundings.

Several anomaly detection techniques have also been applied to images, *e.g.* satellite imagery, spectroscopy, mammographic image analysis and video surveillance [6, 13-15]. Outliers in images are usually identified as either single anomalous points/pixels or as entire sub-regions, and they are detected by analyzing several image attributes such as color, lightness and texture.

In our work we combine the two previously cited classes of anomaly detection methods (text and images), to find outliers on the basis of both the quality of the textual information provided by users and the overall quality of the images associated with those textual elements.

Assessing the overall quality of natural images is a difficult task, as many image attributes need to be taken into account and the notion of "quality" is often subjective. One of the simplest and more relevant image quality measure is provided by focus measure operators [16], which typically analyze the spectrum of the given natural image to detect the presence of strong edges and determine whether the image has been acquired at the right distance from the camera sensor. Another image quality measure that we consider relevant to determine fake and low quality user-generated product descriptions is the presence of hyperlinks within images attached to the textual content, as those hyperlinks usually lead to malicious third party websites.

Since it is difficult and time expensive to collect a decent amount of fake, low quality or badly written user-generated product description, our method is closely related to all those One Class Classification works that try to classify positive cases without exploiting information collected from negative ones [17].

Many works from One Class Classification research area are focused on text classification, e.g. Liu et al. [18] treat the problem of building text classifiers using positive and unlabeled examples, using a biased formulation of Support Vector Machine (SVM) [12], obtaining state-of-the-art results; Manevitz and Yousef [19] compare different One Class Classification models

in the context of Information Retrieval, showing that SVM and Neural Networks obtain the highest detection rates.

In our work, we exploit information extracted from high quality user-generated product descriptions to train a one class SVM classifier that detects anomalous user-generated product descriptions by jointly analyzing different type of data.

3. PROPOSED METHOD

The proposed approach is presented in this section: we define a set of features describing the goodness of both textual descriptions (Section 3.1) and related images (Section 3.2), and those textual and visual features are used to train a one class SVM model (Section 3.3).

3.1. *Textual features*

We analyze the textual information associated with each product description focusing on the aspects that, in our application context, usually characterize a high quality user-generated product description. In the following paragraphs, we illustrate the aspects we have taken into account, pointing out the intuitions behind each of them.

Description Length. When focusing on finding abnormal/ anomalous product descriptions, one of the first feature that needs to be taken into account is the length of the description in terms of number of text characters. The assumption behind this feature is that both an excessively long or short product description should rise a warning. In fact, in our application context, most expert users tend to provide descriptions that have similar lengths, and thus the variance in terms of number of characters between different high quality product descriptions is usually small. Using this numerical feature, our machine learning model learns the distribution of lengths from the high quality product descriptions in the positive training set, and spots possible outliers during the generalization phase.

Description Language. In our application context, the language used to write the product description is particularly relevant, as high quality product descriptions should always be written using the platform main idiom. To identify the language used in a description we rely on the Language Detection Library for Java [20], which not only detects the main language used in the processed document, but also provides a list of percentages of use of all the other languages found in the same document.[2] We exploit this feature to describe the language of a document as the percentage of use of the main platform idiom in the document itself; the idea behind this choice is that, although some technical words in a document may belong to different languages (product name, seller's contact information, etc.), the platform main language should have a high usage percentage rate.

Description Keywords. In the dataset we collected, and more generally in online marketplaces, it is extremely common to find, associated with each product description, a set of special words, called *tags*, that typically represent a list of keywords of the description itself. In our work, we use those *tags* to compute an index of consistency of the textual information provided by users. In details, for each document, we measure the percentage of *tag* words that appear in the text description. The intuition behind this consistency index is that fake or low quality product descriptions usually have either no *tags* or random and meaningless *tags*.

Presence of Hyperlinks. The presence of hyperlinks strongly characterizes fake and low quality product descriptions. An external reference or a private email addresses usually redirects the user to a third party competitor/advertisement website and should not be tolerated unless most of the other product description quality measures (length, language, *tags*, image quality, etc.) are consistent with those extracted from high quality genuine product descriptions; in which case it may be possible that the hyperlink simply redirects to a media content, e.g. video,

[2] http://code.google.com/p/language-detection/

that cannot be directly attached to the product description. We detect the presence of both email addresses and website URLs in product descriptions using a set of regex-based rules, and, for each document, we use the number of external hyperlinks in the associated product description as input feature to our one-class SVM model.

(a) focused (b) unfocused

Figure 1. *Examples of focused (a) and unfocused (b) images of commercial products correctly discriminated by Variance of Laplacian [21] focus measure operator.*

3.2. *Visual features*

The images associated with each product description are a valuable source of information and may help in determining the quality and genuineness of the description itself. The main intuition is that, in high quality documents, a product image should clearly shows the object described in the textual description, without reporting further irrelevant or malicious information.

Even though classifying the content of natural images is difficult and expensive in terms of both computational power and time required to extract visual features and train the classification model, it is still possible to define some simpler image quality measures that are relevant to our application context. In particular, for each user-generated image, we take into account both its focusness level, and the presence of hyperlinks within the image itself. Unfocused images are usually associated with low-quality product descriptions, while images containing hyperlinks

leading to third party websites are typically associated with fake or low quality product descriptions.

The two previously cited image quality indexes used in our work are briefly described in the following paragraphs.

Focusness. In order to determine the best focus measure operator for our application context, we compared most of the focus indexes proposed in literature throughout the last decade [16] over focused images extracted from our set of high quality genuine product descriptions. In our experiments, Variance of Laplacian (LAPV) [21] provided the best results both in terms of computational complexity and discriminative power, as it almost always associated high focus values to our positive focused images. Figure 1 shows some examples of focused and unfocused images that LAPV correctly processes (high and low focus values for focused and unfocused images respectively). In our pipeline, for each processed document, the average output of LAPV for all the images associated with the document is provided as input feature to our one-class SVM model. As previously stated, a high average focusness value should denote a finely crafted user-generated product description.

(a) allowed text (b) external references

Figure 2. *Examples of images of commercial products containing allowed text words (a) and malicious hyperlinks/email addresses (b)*

Presence of Visual Hyperlinks. As described in Section 3.1, hyperlinks typically appear in low quality product descriptions, as they are used to redirect users to third party websites. In some cases, malicious users are aware that their product descriptions

may be penalized whenever they contain external references, therefore, instead of injecting hyperlinks into the textual content, they embed them into the images of products. To detect those image hyperlinks, we exploit a properly trained version of the Tesseract Optical Character Recognition Engine [22]. It is important to note that not all the text found in an image should be considered anomalous, e.g. Figure 2 shows several examples of images containing allowed text elements and abnormal or malicious text components. For this reason, once and if text among an image is detected, we apply the same regex-based rules of Section 3.1 to determine whether it represents a hyperlink/email address or not. Equivalently to our *Presence of Hyperlinks* textual feature, for each product image, the number of hyperlinks and email addresses found in the image are used as input feature to our classifier.

3.3. *Proposed model*

As described in Section 2, we approach the problem of finding fake or low quality commercial product descriptions as an anomaly detection problem due to the fact that, in our application context, the amount of available negative data is not large enough to train a binary classifier.

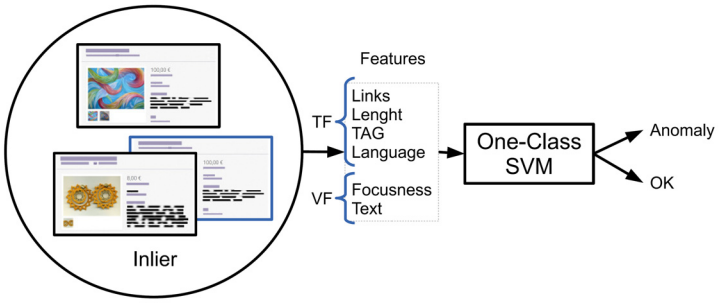


Figure 3. *Visual representation of the training pipeline for the proposed method. The one-class Support Vector Machine model is trained using both textual and visual features (TF and VF respectively) extracted from high quality genuine commercial product descriptions (*inliers*)*

Similarly to other works in literature, we employ a one-class SVM model to infer the behavioral traits of expert users from a dataset composed of high quality genuine product descriptions (described in Section 4.1), using the set of textual and visual features defined in Sections 3.1 and 3.2 respectively. The motivation behind the use of a one-class SVM is that, in anomaly detection, such machine learning model usually lead to optimal results with minimal tuning effort [19].

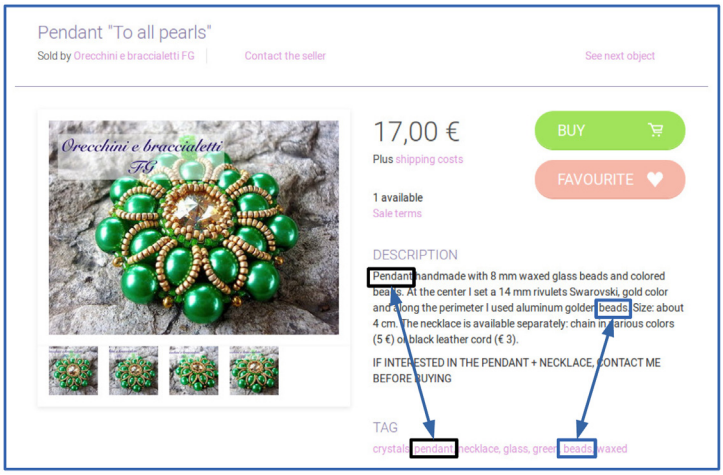
The training phase of the proposed method is summarized in Figure 3. Each positive *inlier* document, typically composed of a textual description and one or more images, is represented as a vector, whose components are the normalized values of the textual TF and visual VF features described in Sections 3.1 and 3.2.

The one-class SVM model is trained using only positive examples, as in classical anomaly detection approaches. When training using a decent amount of significant positive data, we expect the classifer to automatically detect whenever a new product description, that has never been seen during the training phase, contains some anomalous components that substantially differ from those of high quality genuine product descriptions.

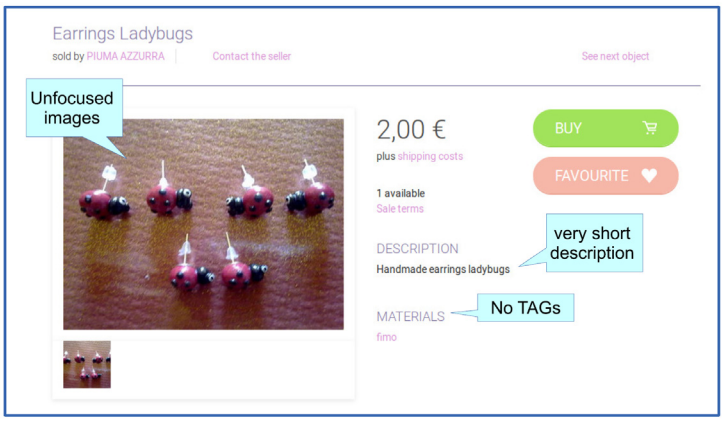
Throughout our experiments (see Section 4), we prove that the proposed set of textual and visual features is exhaustive for the proposed task and that the pipeline described above performs well, achieving high detection rates on the collected dataset.

4. EXPERIMENTS

In this section, we provide an experimental evaluation of the components described in Section 3, and we describe the dataset used in our experiments.



(a) Positive



(b) Negative

Figure 4. *Positive and negative samples extracted from the dataset used in our experiments. (a) A high quality genuine commercial product description containing: an exhaustive textual description,* tags*, a well focused image,* etc. *(b) An artificially generated low quality commercial product description obtained by injecting random anomalies into a high quality user-generated document.*

4.1. *Dataset*

We gather a dataset composed of 41635 documents from a website specialized on selling handmade products. Each insertion is user-generated and composed of a textual description, a set of keywords or tags and one or more images representing the item. All the collected high-quality descriptions documents were manually validated by the administrators of the website.

As in other anomaly detection works [5, 7], the set of possible anomalous descriptions has been artificially built by randomly injecting different kinds of anomalies into high quality genuine user-generated documents, trying to cover a large set of possible anomalous behaviors.

More in detail, we artificially create a total of 1000 negative documents:

- 100 anomalous documents for each type of anomaly that our classier may detect: anomalous length, anomalous language, missing/wrong tags, presence of hyperlinks/email addresses, unfocused product images, and images containing hyperlinks.
- 400 anomalous documents containing two or more types of randomly generated anomalies.

Figure 4 shows a positive high quality product description and an artificially created negative product description containing more than one anomaly. In the positive document of Figure 4 (a), the image of the product is well focused and contains allowed text. On the other hand, in the negative example of Figure 4 (b), the image is unfocused and the item is not clearly displayed. Moreover, the negative example contains a very short textual description and no tags, while in the positive document the object is exhaustively described and two tags are correctly matched in the textual description.

In our experiments, 80% of the high quality genuine documents are used for training, while both the remaining 20% and the artificially created anomalous descriptions are used for testing.

4.2. *Results*

In our experiments, we use the implementation of one-class SVM with RBF kernel provided by the LibSVM Library [12]. The one-class SVM model is trained using 33308 randomly sampled positive documents from the dataset described in Section 4.1. As shown in Figure 3, each document is processed as a vector whose components are the values of the textual and visual features introduced in Sections 3.1 and 3.2 respectively.

As in other works in literature [23, 24], when the number of available positive and negative test documents is strongly unbalanced, it is a common practice to split the set of positive documents into multiple sets having the size of the set of negative documents, and then average the results obtained over each split to compute the final overall result. In our experiments, we divide the 8327 positive test documents into 8 equal splits, and we build 8 different test sets by adding each split to the set of anomalous documents. Each test set is composed of 1040 positive and 1000 anomalous documents.

The goodness of the proposed system at detecting anomalous commercial product descriptions is measured using the following evaluation metrics:

$$\text{Accuracy} = \frac{\text{anomalies correctly identified} + \text{anomalies wrogly identify}}{\text{total \# of documents}}$$

$$\text{Precision} = \frac{\text{anomalies correctly identified}}{\text{total \# of documents marked as anomalous}}$$

$$\text{Recall} = \frac{\text{anomalies correctly identified}}{\text{total \# of anomalies}}$$

$$\text{F-measure} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}$$

The overall accuracies obtained by the proposed model are presented in Table 1. They are computed as the arithmetic mean of the accuracies obtained by the trained one-class SVM model on the 8 1040=1000 test datasets. It can be observed that some positive documents are classified as anomalous as their vectorial representations lay on the separation boundary between positive

and negative documents learned by the one-class SVM during the training phase.

Table 1. *Confusion matrix and evaluation results*

	Anomalies	Correct
Identified as anomalous	858	94
Identified as correct	142	946

Accuracy	0.88
Precision	0.90
Recall	0.85
F-measure	0.88

To evaluate the capability of the proposed method at detecting different types of anomalies, for each type of anomalous content considered in our work (length, language, *tags*, hyperlinks in text, focusness and hyperlinks in image) a detection rate has been calculated. For every anomaly type, the respective detection rate is defined as follow:

$$\text{Detection Rate } (\%) = \frac{\text{anomalies correctly identified}}{\text{total } \# \text{ of anomalies}}$$

Results are provided in Table 2. Commercial product descriptions containing more than one type of anomalous behavior are almost always detected, while descriptions that have been injected with single anomalous behaviors are harder to detect, especially the ones containing non matching *tags*. Unsurprisingly, descriptions injected with anomalous visual behaviors (lack of focus and presence of hyperlinks in images) are easily detected, highlighting the importance of considering both visual and textual informations when searching for fake or low quality user-generated content.

Table 2. *Detection rate with different types of anomalies*

Anomalies	Detection Rate
Anomalous length	39%
Anomalous language	79%
Missing/wrong tags	43%
Presence of hyperlinks	99%
Unfocused images	97%
Images containing hyperlinks	98%
$\geq$ 2 random anomalies	97%
Average	78%

In terms of computational complexity, the proposed model requires on average roughly 220 ms to process the textual information associated with each textual description, and a total of roughly 350 ms to process all the visual elements associated with the same description (67 ms for LAPV focus measure operator and 287ms for Tesseract OCR hyperlink extraction).

5. CONCLUSION

A novel anomaly detection method for identifying fake and low quality usergenerated commercial product descriptions has been proposed, it exploits features extracted from both textual and media content to obtain excellent detection rates. Thanks to the use of both a focus measure operator that computes the overall quality level of images, and a text localization/recognition system that identifies hyperlinks and email addresses within images, the proposed system effectively recognizes unusual product descriptions that cannot be detected when exclusively analyzing their textual content. The use of a compact set of highly discriminative features enables our system to process user-generated commercial product descriptions in real time, marking the ones that potentially contain suspicious content for further manual inspection.

REFERENCES

1. Hodge, V. & Austin, J. 2004. A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22, 85-126.

2. Chandola, V., Banerjee, A. & Kumar, V. 2009. Anomaly detection: A survey. *ACM Computing Surveys*, 41, 1-58.
3. Pasha, M. Z. & Umesh, N. 2012. Outlier detection: Applications and techniques. *International Journal of Computer Science*, 9, 307-323.
4. Brause, R., Langsdorf, T. & Hepp, M. 1999. Neural data mining for credit card fraud detection. In *IEEE International Conference on Tools with Artificial Intelligence*.
5. Fan, W., Miller, M., Stolfo, S. J., Lee, W. & Chan, P. K. 2001. Using artificial anomalies to detect unknown and known network intrusions. In *IEEE International Conference on Data Mining.*
6. Spence, C., Parra, L. & Sajda, P. 2001. Detection, synthesis and compression in mammographic image analysis with a hierarchical image probability model. In *IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*.
7. Guthrie, D., Guthrie, L., Allison, B. & Wilks, Y. 2007. Unsupervised anomaly detection. In *International Joint Conferences on Artificial Intelligence*.
8. David Guthrie, L.G. & Wilks, Y. 2008. An unsupervised approach for the detection of outliers in corpora. In *International Conference on Language Resources and Evaluation*.
9. Amogh Mahapatra, N. S. & Srivastava, J. 2012. Contextual anomaly detection in text data. *Algorithms*, 4, 469-489.
10. Blanchard, G., Lee, G. & Scott, C. 2010. Semi-supervised novelty detection. *Journal of Machine Learning Research*, 11, 2973-3009.
11. Baker, D., Hofmann, T., McCallum, A., Yang, Y. 1999. A hierarchical probabilistic model for novelty detection in text. In *International Conference on Machine Learning*.
12. Chang, C. C. & Lin, C. J. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2, 1-27.
13. Theiler, J. & Cai, D. M. 2003. Re-sampling approach for anomaly detection in multispectral images. In *International Society for Optical Engineering*.
14. Chen, D., S.X.H.B. & Su, Q. 2005. Simultaneous wavelength selection and outlier detection in multivariate regression of near-infrared spectra. *Analytical Sciences*, 21, 161-167.
15. Pokrajac, D. 2007. Incremental local outlier detection for data streams. In *IEEE Symposium on Computational Intelligence and Data Mining*.
16. Pertuz, S., Puig, D. & Garcia, M.A. 2013. Analysis of focus measure operators for shape-from-focus. *Pattern Recognition*, 46, 1415-1432.

17. Khan, S.S. & Madden, M. G. 2010. A survey of recent trends in one class classification. In *Irish Conference on Artificial Intelligence and Cognitive Science*.
18. Liu, B., Dai, Y., Li, X., Lee, W.S. & Yu, P. S. 2003. Building text classifiers using positive and unlabeled examples. In *International Conference on Data Mining*.
19. Manevitz, L. M. & Yousef, M. 2002. One-class svms for document classification. *Journal of Machine Learning Research*, 2, 139-154.
20. Shuyo, N. 2010. Language detection library for java (2010) Software available at http://code.google.com/p/language-detection/.
21. Pacheco, J. P., Cristobal, G., Martinez, J. C. & Valdivia, J. F. 2000. Diatom auto focusing in brightfield microscopy: A comparative study. In *IEEE International Conference on Pattern Recognition*.
22. Smith, R. 2007. An overview of the tesseract ocr engine. In *International Conference on Document Analysis and Recognition*.
23. Chawla, N. 2005. Data mining for imbalanced datasets: An overview. In *Data Mining and Knowledge Discovery Handbook* (pp. 853-867).
24. Bekkar, M., Djemaa, H. K. & Alitouche, T. A. 2013. Evaluation measures for models assessment over imbalanced data sets. *Journal of Information Engineering and Applications*, 3, 27-38.

LUCIA NOCE
UNIVERSITY OF INSUBRIA,
DEPARTMENT OF THEORETICAL AND APPLIED SCIENCE,
VIA MAZZINI, 5, 21100 VARESE, ITALY.
E-MAIL: <LUCIA.NOCE@UNINSUBRIA.IT>

IGNAZIO GALLO
UNIVERSITY OF INSUBRIA,
DEPARTMENT OF THEORETICAL AND APPLIED SCIENCE,
VIA MAZZINI, 5, 21100 VARESE, ITALY.
E-MAIL: <IGNAZIO.GALLO@UNINSUBRIA.IT>

ALESSANDRO ZAMBERLETTI
UNIVERSITY OF INSUBRIA,
DEPARTMENT OF THEORETICAL AND APPLIED SCIENCE,
VIA MAZZINI, 5, 21100 VARESE, ITALY.
E-MAIL: <A.ZAMBERLETTI@UNINSUBRIA.IT>

International Journal of Computational Linguistics and Applications, Vol. 6, No. 2, 2015, pp. 177–192
Received 22/06/2015, Accepted 24/07/2015, Final 23/09/2015.
ISSN 0976-0962, http://ijcla.bahripublications.com

A Text Mining Library for Biodiversity Literature in Spanish

JUAN M. BARRIOS
ALEJANDRO MOLINA
RAUL SIERRA-ALCOCER
ENRIQUE-DANIEL
ZENTENO-JIMENEZ
The National Commission for Knowledge and Use of Biodiversity (CONABIO)

ABSTRACT

Biodiversity represents a great ecological, economic and aesthetic heritage to the world. Most of the knowledge about this heritage could be found in thousands of documents that describe valuable information obtained over centuries.

Projects which try to gather and structure all this information, even for very specific topics, may take years. In addition to this, keeping a project updated is difficult because new knowledge is continuously being published. Therefore, there is a necessity to use automatic methods to extract relevant information efficiently. In this article we describe the first stage of a software project, that aims to build a complete library to apply Natural Language Processing techniques on documents about biodiversity in Spanish.

1. INTRODUCTION

This project is part of a large and permanent effort at the National Commission for Knowledge and Understanding of Biodiversity (CONABIO) to gather information about biodiversity in Mexico. The mission of the CONABIO is to promote, coordinate, support and carry out activities aimed at the knowledge of biological diversity in Mexico, and its preservation

for the benefit of society. As part of this mission, CONABIO is invested in creating and publishing knowledge databases about Mexican biodiversity. Up to know, most of the efforts in this direction have been carried out through traditional literature review. Every new project represents a significant challenge because it requires to select and organize thousands of potentially relevant documents to a new topic and then to extract the information on those documents and structure it on databases.

Natural Language Processing techniques offer an opportunity to automatize some of the tasks involved in these projects, saving many person-hours and increasing effciency by orders of magnitude.

In this article we present the first delivery of a Text Mining library focused on the extraction of information about biodiversity from documents in Spanish. This is the first stage of a software project that will go from extraction of plain UTF-8 text from PDF/OCRed files to the automatic extraction of fragments about uses of biodiversity in Mexico.

The progress made during the first phase includes the following features:

- OCR/parsing of PDF files,
- parsing correction,
- sentence segmentation,
- traditional species uses extraction,
- indexing of named entities,
- efficient Global Names Recognition and Discovery service call,
- extraction based in lexical patterns,
- and multiprocessing.

The tools in this first phase will enable us to develop more complex modules. For instance, we are currently working in a data set to induce models for Named Entity Recognition focused on species. The development of this library is an open initiative licensing under GPLv2[1], and can be downloaded from the repository <https://bitbucket.org/conabio_cmd/text-mining>.

[1] http://www.gnu.org/licenses/old-licenses/gpl-2.0.en.html

2. STATE OF THE ART

Some NLP tasks require to be adapted for application in biology. As a consequence, new NLP challenges have emerged thanks to the interaction with biosciences data. For instance, the Biodiversity Heritage Library is in the process of digitizing 600000 pages of text a month, making them available as pdf image les and OCR text files.[2] However, biodiversity literature can be especially difficult to OCR and the current rate of digitization prohibits manual correction of these errors. Proposed solutions include components of crowd-sourcing manual corrections for automated corrections [1].

Aside from text correction, domain terms extraction is also a common topic concerning NLP applied to biology. Named Entity Recognition efforts have been oriented to detect species names (taxon) using mainly two approaches: lexicon based (dictionaries) and machine learning based. Lexicon based approaches focus on finding words that are contained in dictionaries previously given to the computer. An example is Linnaeus, designed specifically for identifying taxonomic names in biomedical literature using pattern matching [2]. Taxon Finder detects scientific names by comparing the name to several lists [3]. Taxon Grab uses a combination of nomenclatural rules and dictionaries of non-taxonomic English terms [4]. Supervised machine learning approaches rely on providing substantial training examples to a system that would reproduce a specific task. In the case of taxon name detection, the letter combinations within the names as well as the context are helpful to recognize scientific names. NetiNeti, a supervised learning algorithm, based on Bayes conditional probability, uses these features [5]. NetiNeti may learn, for instance, that a word with the first letter capitalized and ending with “es” is probably a taxon name, even though that word has never appeared in training examples.

Concerning Information Extraction, very interesting applications has recently been proposed. In [6], authors describe an algorithm to learn rules to extract leaf properties from plant

[2] www.biodiversitylibrary.org

descriptions. Another example is described in [7], where authors try to match patterns relating proteins (X activates Y, Y is activated by X, Y was activated by X, etc.).

Nevertheless, the majority of ready-to-use tools are only for English; this is the main reason to start a new project for Spanish and using Mexican literature.

3. PDF PARSING AND OCR

A common technical difficulty when working with PDF files is that those files may not have a text layer, in that case, it is necessary to apply optical character recognition (OCR). Sometimes, a PDF file has a text layer, but it does not have the permissions to extract it automatically.

For all this, we have developed a controller based on three different PDF parsers to obtain plain text[3]: Apache PDFBox, Apache Tika[4] and pdftotext[5]. Thus, we have managed to get plain text in most of cases. However, PDF parsers not always offer good results. Perhaps given the complexity of format in names in Latin, tabular information, bibliographical citations, varied typography and text columns; all this being quite common in specialized texts.

Using the library, it is possible to apply all of the PDF/OCR parsers at once. For instance, from the command line, we would do:

```
# txtm.py PDFparserController --infile f.pdf --
parser_type all
```

The file `f.pdf` will be parsed and for each parser a directory containing the extracted raw text will be created. This is useful to evaluate the quality of different outputs.

[3] PDF parsers require Java Runtime Environment (JRE).
[4] © The Apache Software Foundation
[5] © The Poppler Developers

4. Correction of Sentences Frontiers, Abbreviations and Hyphens

Since the text obtained from the parsing of PDFs is aligned with the print view, it is necessary to make corrections to reconstruct sentences. Example 1 shows a fragment of text extracted in which line breaks are found to fit the print format. This kind of text segmentation is useless to apply even the most basic techniques of NLP. For instance, the detection of a specific syntactic pattern in a badly splited sentence is impossible. Other sources of error in text segmentation are hyphened words and abbreviations, which may be confused with end of sentence punctuation.

> Example 1. En el transcurso de la elaboración de esta obra se fueron sumando participantes, de manera que a su conclusión cuenta con 79 colaboradores pertenecientes a 19 instituciones tanto académicas, como gubernamentales y no gubernamentales (cuadro 1). El Estudio está conformado por...

It has been necessary to include a specialized module to correct hyphened words and to detect the borders of sentences. Hyphen correction is based on regular expressions; while sentence segmentation is based on supervised learning. Using thousands of sentences manually annotated from a general corpus in Mexican Spanish; we have obtained a 90% precise segmentation on general texts. Both, training and evaluation of sentence segmentation are done using a wrapper of apache OpenNLP Sentence Detector.[6]

Although some segmentation problems are mitigated, errors still persist since there are countless abbreviations in the biological domain that are not included in our general corpus examples; therefore may not be learned by our segmentation model. In this regard, we have integrated a segmentation model using an specialized corpus from the domain and focused on examples of abbreviations and citations.

---

[6] http://opennlp.apache.org/documentation/manual/opennlp.html

Using the library, it is possible to apply the complete correction. From the command line, we would do:

```
# txtm.py PreprocessingController \
--infile f.txt \
--opennlp_bin <path_to_opennlp_bin> \
--opennlp_mod        /path/txtmining/txtmining/resources/
models/es-iula.bin \
--correction_type all
```

5. COMMON AND SCIENTIFIC NAMES OF MEXICAN SPECIES

Identifying species names in biodiversity literature is critical for a number of applications in data mining. At the current stage of the project, this task is tackled by using a lexicon-based approach that follows some ideas exposed in [2] adapted to Mexican species. We considered also the basic rules of scientific names writing mentioned in [8]. The lexicon lookup strategy scans a given input document, looking for terms that match (1) a word from the genera list, e.g. *Abeis*; (2) a word from the genera list followed by a word from the species list, e.g *Abeis mexicana*; (3) a word from the genera list followed by a specific abbreviation of species, e.g. *Abeis* (*sp*., *ssp*., *subsp*., *nov*.); (4) the abbreviation of a genus followed by a word from the species list or a specific abbreviation of species, e.g. *A. mexicana*; or (5) a common name from the list of common names, e.g. *Abeto de Vejar*.

We have included, in our development list, around 1000 genera, 2600 species and 13000 common names. However, new names or slightly bad written names will not be found, no matter how exhaustive are the list. For this reason, we have integrated the tool described in section 6 for name discovery and resolution.

Dealing with species common names is far more challenging. The lexicon lookup strategy suffers from important problems: (1) when names are homographs of other nouns (like *bandera*, *baraja*); (2) when names are homographs of common use words (like *ni*, *mis*, *ya*, *lo*); and (3) when short names match long ones too (like *barba* in *barba de chivo* or *palo* in *palo de agua*). The third complication could be solved using length-sorted lexicons. However, the two first problems need more accurate algorithms

to be detected and disambiguated. In future versions of the library, this feature will be added. The current version allows to integrate easily custom lists of names, basically by adding a file. The command to create an index of names contained in a file using a lexicon is:

```
# python txtm.py ReIndexController \
--infile f.txt --regexp_file <path_to_regexp_file>
```

6. GLOBAL NAMES RECOGNITION AND DISCOVERY SERVICE

The Global Names Recognition and Discovery (GNRD) service is a tool to recognize scientific names based on TaxonFinder [3] and NetiNeti [5] names discovery engines. Found names are optionally resolved against a number of resources.[7]

TaxonFinder detects only scientific names. Given a text, it will scan through the contents and it will use a lexicon-based approach to identify which words and strings are Latin scientific organism names. It also detects names at all ranks, including species, genus and subspecies but does not detect common names.

NetiNeti detects scientific names using machine learning. The system estimates the probability of a label (whether a name is scientific or not) by given a candidate string along with its contextual information.

The final response of GNRD service will combine the advantages of both engines. However, the language of incoming content is determined using unsupervised language detection. If the language found is other than English, only TaxonFinder is used. Therefore, the resolver best performance is expected to be for English. Nevertheless, it also shows enough accuracy for Spanish; considering that names can be optionally resolved by using some other certified resources. Our solution is only to consider certified names to be included in the final answer. It should be noted that many Mexican species are not registered in such resources, especially endemic organisms.

[7] http://resolver.globalnames.org/data sources!

One drawback using GNRD service is that it experiences long network delays when many large documents are trying to be resolved. Furthermore, after a long delay it is possible to receive empty answers or error codes. Consequently, we have developed an efficient caller which first divides the texts in lots, and for each one, it sends a request to the resolver. Finally, the responses are merged to have one single index. The main advantage of this strategy is that if one request fails, only a part of the index will be lost while the other lots could have non-empty responses. Moreover, each request could be asynchronous using a task manager library.

The command to create the index of names of a file using Global Names is:

```
# python txtm.py GnIndexController --infile f.txt
```

7. Automatic Extraction of uses of Biodiversity in Mexico

In terminology extraction, some methods based in syntactic patterns in Spanish have been proposed to detect functional definitions into specialized texts [9]. A functional definition is when a term *T* is associated with some specific use *U* by a syntactic pattern called functional verbal predication. This verbal predication is simply a verbal form generally used to describe *T*'s uses.

In this first version of the library, we have developed an extractor that identifies fragments about the use of Mexican species with the help of patterns of type "*T* + {fused as} + U". Example 2 presents the term *Tunillo*, the common name of a Mexican cactus; and the functional verbal predication *used as*. It is important to note that these patterns should be detected simultaneously in the same fragment and this is the reason why the segmentation by sentences presented in Section 4 is necessary.

In addition to common names, scientific names and verbal functional predications, there are other elements of interest that have been integrated such as parts of animals and plants, names in native languages, names of objects for domestic use, hunting and fishing instruments, names of conditions in indigenous

communities, and names of therapeutic practices, among others. All these resources were provided by experts at CONABIO.

> Example 2. Tunillo (Stenocereus treleasei):
> *se come y se vende el fruto, hay quien hace juguetes con los tallos, y se pega una parte del tallo detrás de las orejas cuando hay paperas, se usa también como cerco vivo.*[8]

Syntactic based methods, however, present two major disadvantages: (1) they may extract false positives, and (2) they cannot extract fragments that do not contain verbal patterns. Consequently, we plan to annotate and validate manually the fragments extracted on this stage in order to generate a dataset to train supervised learners that can extract this information from fragments that do not present all the syntactic elements.

We can extract this sort of fragments with our library using the following command:

```
  # python txtm.py FragmentController --infile f.txt \
      --regex_dir <path_to_regexp_dir>

> {"documentName": "f.txt",
    "lineNumber": 1,
    "documentFragment": "Tunillo ...",
    "commonName": [{"instance": "Tunillo","offsetStart": 1,
      "offsetEnd": 8},...],
    "sciName": [{"instance": "Stenocereus treleasei", "offsetStart": 10
      "offsetEnd": 31},...],
    "parts": [{"instance": "fruto","offsetStart": 55,
      "offsetEnd": 60},
      ...],
    "functionalVerb": [{"instance": "usa","offsetStart": 176,
      "offsetEnd": 179}],
    "handcraft": [{"instance": "juguetes","offsetStart": 77,
      "offsetEnd": 85},...]
}
```

[8] An approximated translation will be: Tunillo (Stenocereus treleasei): the fruit is eatable and sold; some others make toys using its stems, or stick a piece of its stalk behind ears when they have mumps. It is also used as a hedge.

8. PRELIMINARY RESULTS AND DISCUSSION

8.1. *Sentence segmentation*

In Table 1, we show the improvement in sentence frontiers detection after using specific domain corpora as well as a specialized dictionary of abbreviations to train a maximum entropy model. The best model was trained with 4.4G words (185.1M sentences) from the Environment and Medicine documents of the CTIULA[9] Technical Corpus [10] using 10,000 iterations and cuto equals to 4 as parameters.

Sentence frontiers detection is crucial to the rest of the tasks because many of them depend on the quality of text segmentation, as we mentioned above. For instance, in species names detection, it could be an important feature the fact that other taxon names appear in the same sentence.

Table 1. *Comparison of trained models for sentence frontiers detection*

	Precision	Recall	F-measure
General Spanish	0.6241	0.7011	0.6604
CT-IULA (environment & medicine)	0.8333	0.8897	0.8606
CT-IULA + ad hoc Abbreviations	0.9211	0.9003	0.9106

8.2. *Taxon names detection*

A crucial task for text mining for biodiversity literature is to find scientific names of species. In section 5, we have described an experimental scientific names indexer based on regular expressions (RegExp) containing around 1000 genera and 2600 species names of Mexican trees. Later in section 6, we have described Taxon Finder [3] and NetiNeti [5]. The former based on lexicons and the later based on machine learning methods.

Table 2 presents the results for the evaluation of different methods for scientific names detection: a tree specialized Regex, Taxon Finder and a NetiNeti model trained for biodiversity literature in Spanish. For this last, we created a dataset to train NetiNeti models in Spanish, the best NetiNeti model obtained

[9] http://www.iula.upf.edu/corpus/corpusuk.htm

with literature in Spanish is what we call SpaNeti.We also show the results using the default train parameters for English NetiNeti to point out the improvement after using texts in Spanish to train the model. We have evaluated all of the methods using the same, manually annotated, text about Mexican trees [11]. Evaluation is composed of three sub-tasks: to seek out one-word taxons (Monomial), to seek out two-word taxons (Binomial) and to seek out any length names (Any Taxon).

Table 2. *Comparison of tools for taxon names detection in a text about trees*

	Monomial		Binomial		Any Taxon	
	Precision	Recall	Precision	Recall	Precision	Recall
RegExp (ad hoc trees)	1.0000	0.6821	1.0000	0.5748	0.9117	0.4033
Taxon Finder	0.8754	0.9014	0.9352	0.8019	0.8339	0.8587
SpaNeti (Spanish model)	0.9120	0.6171	0.9545	0.7608	0.8379	0.5669
NetiNeti (Default model)	0.4383	0.6542	0.6494	0.7874	0.3872	0.5780

As can be expected, extracting names based on RegExp is limited to the dictionary employed which is relfected on the low recall. TaxonFinder presents a more stable result than any other method used but there is an improvement on the extraction if we use SpaNeti. Therefore we propose that the best strategy for scientific names detection is to combine both methods.

Below we are going to discuss more thoroughly what are the advantages and disadvantages of each method.

Regex Using regular expressions we obtain a perfect precision score of 1.0 for Monomial and Binomial names tasks but not for Any Taxon because in this last task we have considered complete names as the correct answer. In consequence, the name “Pinus pseudostrobus” found by Regex is penalized against the more specific, say longer name “Pinus pseudostrobus var. oaxacana”. Indeed, our experimental Regex had not rules about names whose length is greater than two words. After our experiments, we found out that trying to capture all possible taxon name cases in one single expression could be very challenging. Hence, not the easiest strategy.

The second observation is that although the test text is about trees, it is common to find names of species other than trees. Therefore, the recall is penalized.

However, the most significant drawback of Regex is that it does not recognize subtle differences in species names. It does not match taxon names if they are not written exactly as they appear in the regular expression. For instance, if an author has named "magnolifolia" (instead of "magnoliifolia") the taxon will not be retrieved. In example 3 "Q. glaucoides" and "Quercus" are detected but "Q. magnolifolia" is not.

> *Example 3. La especie con los valores más altos de importancia es Q. magnolifolia (185.29) esta especie presenta los valores más altos de densidad (60), frecuencia (50) y cobertura (75), los otros encinos presentes en este sitio presentan valores bajos de importancia Q. glaucoides (26) Quercus 346 (24) y Quercus 347 (20), sin embargo es el único sitio de la zona intermedia con más de una especie de encinos, también es el unico sitio en el que se encontró al Timbre con valores de importancia moderados (44.27) así como de frecuencia (20) y cobertura (11) (Cuadro 7) (Figura 12).*

Taxon Finder This method obtained the best scores (over 80%) of recall for all tasks. The advantage of Taxon Finder is that it uses dictionaries from more than two levels in the taxonomic hierarchy, and rules to detect inferior levels like subspecies, race and variety. This enables it to identify long names like "Pinus pseudostrobus var. oaxacana". Taxon Finder, however, like the regex method does not recognize names with subtle differences in writing. In consequence, the same omission in Example 3 is expected.

An interesting aspect of results from Taxon Finder is that sometimes, names of rivers or locations are confused with species. In example 4, "Atoyac" and "Bejuco" are retrieved because the former is included in the genera dictionary and the second is confused with "Bejuco pendulus Loe." (a synonym of "Hippocratea volubilis L."). Similar phenomena occurs for "Calera", "Jarilla" and "Huerta".

Example 4. La zona forma parte de la región hidrológica Río Atoyac (RH-20), destaca el Río Molino que nace del Río la Catrina, Río Obscuro y Río Bejuco a una altitud de 2900 msnm en las faldas de la ladera sur de la Peña Boluda o Peña de San Felipe, que por su caudal es la principal corriente que llega a la población (INEGI, 2006)

SpaNeti To get more flexibility in the detection of scientific names we can use machine learning techniques, NetiNeti is a tool designed for this purpose, but is trained for English. SpaNeti, our best model trained with literature in Spanish obtained the following results.

SpaNeti gives many false positives with words starting with uppercase and ending in {“a”, “s”} like the following examples, some of them proper nouns: “Resulta”, “Catarina”, “Primaria”, “Secundaria”, “Frecuencia”, “Cobertura”, “Toda”, “Piedra”, “Esta”, “Oaxaca”, “Biznaga”, “Pingüïca”, “Bretónica”, “Mata”, “Higuerilla”, “Jarilla”, “Salvia”, “Ambas”, “Naturales”, “Medicinales”, “Algunas”, “Ornamentales”, “Centrales”, “Comunales”.

Since Spanish is a Romance Language it is natural that some heuristics that work to distinguish scientific names in texts in English do not work for texts in Spanish. We believe that the problem described above is due to some rules that NetiNeti uses for the ending letters that increase the probabilities for words ending in {“a”, “I”, “s”, “m”}.

But the strength of SpaNeti is that it is capable of finding names that do not have the orthography of the lexicon, these are some examples that SpaNeti does detect: “Coryphanta retusa” (Coryphantha retusa), “Quercus magnolifolia” (Quercus magnoliifolia), “Abies hickeli” (Abies hickelii).

9. Conclusions and Future Work

We have presented in this article the first version of a long term project that aims to develop tools for extracting information from biodiversity literature in Spanish. The need for this system was identified from the lack of text mining tools for Life Science

literature in Spanish. We have identified some important tasks that need to be completed in order to start applying formal data mining techniques for this purpose, and this first delivery of our Text Mining Library for Biodiversity Literature in Spanish is an effort to compile a set of tools that facilitate the preprocesing tasks that many text mining projects will require before starting to apply NLP techniques. We have also included some basic capabilities that may help finding information about biodiversity, in particular about Mexican trees.

In this first stage we have dealt with extraction of text from PDF files, correction of OCR, taxon recognition and extraction of fragments that are likely to have information about traditional uses of Mexican species. We have shown the advantages of training a model to correct sentences in the specific domain of biodiversity and we have discussed our results in Named Entities Recognition focused on species names.

There are many improvements and lines of work required for this library. To name a few, we need a better model to correct sentences in this specific domain, also entities recognition focused on species names and common names disambiguation is a common task that would be very useful to several projects. As part of the current development, we have been compiling a dataset to apply machine learning techniques. And in future we plan to include the annotation of the corpus to start experiments in other complex tasks like automatic detection of the relationship between a taxon and its aliases. We plan to publish this dataset so that it can serve as a benchmark to other NLP projects in Spanish.

There are many things that need to be done, our hope is that this work will generate interest in the community to contribute in this project full of potential applications.

Figure 1 shows part of the library architecture. In the current version, the controllers layer (in the middle of the figure) enables the access to lower level functionality classes (at the bottom of the figure). The natural way to add functionality to the library is by creating a new branch in the repository and then adding the

unit tests[10] for the new controller (following the test driven development practices mentioned in [12]).

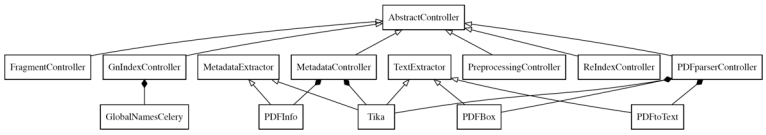


Figure 1. *UML class diagram*

REFERENCES

1. Freeland, C. 2011. Digitization and enhancement of biodiversity literature through ocr, scientific names mapping and crowd sourcing. *BioSystematics Berlin*.
2. Gerner, M., Nenadic, G. & Bergman, C. M. 2010. Linnaeus: A species name identification system for biomedical literature. *BMC bioinformatics*, 11, 85.
3. Leary, P. R., Remsen, D. P., Norton, C. N., Patterson, D. J. & Sarkar, I. N. 2007. ubiorss: tracking taxonomic literature using rss. *Bioinformatics*, 23, 1434-1436.
4. Koning, D., Sarkar, I. N. & Moritz, T. 2005. Taxongrab: Extracting taxonomic names from text. *Biodiversity Informatics*, 2, 79-82.
5. Akella, L. M., Norton, C. N. & Miller, H. 2012. Netineti: Discovery of scientific names from text using machine learning methods. *BMC Bioinformatics*, 13, 211.
6. Tang, X. & Heidorn, P. 2007. Using automatically extracted information in species page retrieval. *Proceedings of TDWG 2007*.
7. Krauthammer, M., Rzhetsky, A., Morozov, P. & Friedman, C. 2000. Using blast for identifying gene and protein names in journal articles. *Gene*, 259, 245-252.
8. Ride, W. D. 1999. International code of zoological nomenclature. *International Trust for Zoological Nomenclatu History Museum*.
9. Sierra, G.E., Alarcón, R., Molina, A. & Aldana, E. 2009. Web explotation for definition extraction. In *IEEE Latin AmericanWeb Congress (LA-WEB'09)* (pp. 217-223), Merida, Mexico.
10. Vivaldi Palatresi, J. 2009. Corpus and exploitation tool: Iulact and bwananet. In *International Conference on Corpus Linguistics*

[10] Test data are provided in the repository.

(CICL 2009) (pp. 224-239), *A survey on corpus-based research*, Universidad de Murcia.
11. Padilla Gómez, E. 2007. Estudio ecológico y etnobotánico de la vegetación del municipio de san pablo etla, oaxaca. Master's thesis, Centro Interdisciplinario de Investigación para el Desarrollo Integral Regional, Unidad Oaxaca. Instituto Politénico Nacional.
12. Astels, D. 2003. Test driven development: A practical guide. Prentice Hall Professional Technical Reference.

JUAN M. BARRIOS
THE NATIONAL COMMISSION FOR KNOWLEDGE AND USE OF BIODIVERSITY (CONABIO)
E-MAIL: <AMOLINA@CONABIO.GOB.MX>

ALEJANDRO MOLINA
THE NATIONAL COMMISSION FOR KNOWLEDGE AND USE OF BIODIVERSITY (CONABIO)

RAUL SIERRA-ALCOCER
THE NATIONAL COMMISSION FOR KNOWLEDGE AND USE OF BIODIVERSITY (CONABIO)

ENRIQUE-DANIEL
THE NATIONAL COMMISSION FOR KNOWLEDGE AND USE OF BIODIVERSITY (CONABIO)

ZENTENO-JIMENEZ
THE NATIONAL COMMISSION FOR KNOWLEDGE AND USE OF BIODIVERSITY (CONABIO)