Ontology-based Technical Text Annotation

François Lévy[†] **Nadi Tomeh**[†] **Yue Ma**[‡]
{francois.levy,nadi.tomeh}@lipn.univ-paris13.fr[†], mayue@tcs.inf.tu-dresden.de[‡]
[†]Université Paris 13, Sorbonne Paris Cité, LIPN, Villetaneuse, France
[‡]Dresden University of Technology, Dresden, Germany

Abstract

Powerful tools could help users explore and maintain domain specific documentations, provided that documents have been semantically annotated. For that, the annotations must be sufficiently specialized and rich, relying on some explicit semantic model, usually an ontology, that represents the semantics of the target domain. In this paper, we learn to annotate biomedical scientific publications with respect to a Gene Regulation Ontology. We devise a two-step approach to annotate semantic events and relations. The first step is recast as a text segmentation and labeling problem and solved using machine translation tools and a CRF, the second as multi-class classification. We evaluate the approach on the BioNLP-GRO benchmark, achieving an average 61% F-measure on the event detection by itself and 50% F-measure on biological relation annotation. This suggests that human annotators can be supported in domain specific semantic annotation tasks. Under different experimental settings, we also conclude some interesting observations: (1) For event detection and compared to classical time-consuming sequence labeling approach, the newly proposed machine translation based method performed equally well but with much less computation resource required. (2) A highly domain specific part of the task, namely proteins and transcription factors detection, is best performed by domain aware tools, which can be used separately as an initial step of the pipeline.

1 Introduction

As is mostly the case with technical documents, biomedical documents, a critical resource for many applications, are usually rich with domain knowledge. Efforts in formalizing biomedical information have resulted in many interesting biomedical ontologies, such as Gene Ontology and SNOMED CT. *Ontology-based semantic annotation* for biomedical documents is necessary to grasp important semantic information, to enhance interoperability among systems, and to allow for semantic search instead of plain text search (Welty and Ide, 1999; Uren et al., 2006; Nazarenko et al., 2011). Furthermore, it provides a platform for consistency checking, decisions support, etc.

Ideal annotation should be accurate, thus requiring intensive knowledge and context awareness, and it should be automatic at the same time, since expert work is time consuming. Many efforts have been made in this field, from named entity recognition (NER) to information extraction (Ciravegna et al., 2004; Kiryakov et al., 2004), both in open domain (Uren et al., 2006; Cucerzan, 2007; Mihalcea and Csomai, 2007) and particular domains (Wang, 2009; Liu et al., 2011). Most cases of NER or information extraction focus on a small set of categories to be annotated, such as *Person*, *Location*, *Organization*, *Misc*, etc. Such a scenario often requires a special vocabulary, and generally benefits much from a limited set of linguistic templates for names or verbs. These restrictions can be widened by linguistic efforts in recognizing relevant forms, but they are the condition of accuracy.

With the increasing importance of ontologies in general or in specific domains[1], annotating a text regarding to a rich ontology has become necessary. For example, the BioNLP ST'11 GENIA challenge

[1]For instance, the OBO site lists 130 biological ontologies. The NASA publishes SWEET, a set of 200 small ontologies dedicated to earth and environment. The ProtegeOntology Library lists around 90 items.

Proceedings of SADAATL 2014, pages 21–30,
Dublin, Ireland, August 24, 2014.

task involved merely 10 concepts and 6 relations, but BioNLP ST'13 GRO task concerns more than 200 concepts and 10 relations. Some ontology-based annotating systems exist and include SemTag (Dill et al., 2003), DBpediaSpotlight (Mendes et al., 2011), Wiki Machine (LiveMemories, 2010). However, each of them is devoted to a particular ontology, for instance, Stanford TAP entity catalog (Guha and McCool, 2003) for SemTag and DBpedia Lexicalization Dataset[2] for DBpediaSpotlight. Hence, these existing systems cannot be directly used to reliably annotate biomedical domain, which is the case of the present work. To this end, the challenge that we focus on is semantic annotation of texts in a particular technical domain with regards to a rather large ontology (a large set of categories), which comes with its technical language and involves uses of concepts or relations that are not named entities. In this kind of use cases, one can get some manual expert annotations, but generally not in large quantity. And one has to learn from them in order to annotate more. This paper experiments on a set of biological texts provided for the BioNLP GRO task[3]. Since our approach is solely data-driven, it can be directly applied to obtain helpful annotation on legal texts governing a particular activity, formalization of specifications and requirement engineering, conformance of permanent services to their defining contracts, etc.

The task at hand is described in section 2, together with the main features of the GRO ontology used in the experiments. We consider here a classical pipeline architecture. The subtasks are recast as machine translation and sequence labeling problems, and standard tools are used to solve them. The first layer is based on domain lexicons and is not our work. Our tools are applied to the detection of relations and events[4]. Section 3 presents experiments, results and comparisons on the annotation of event terms. Section 4 presents experiments in detecting relations and completing event terms with their arguments.

2 A Pipeline Approach to Ontology-Based Text Annotation

The GRO task (Kim et al., 2013) aims to populate the Gene Regulation Ontology (GRO) (Beisswanger et al., 2008) with events and relations identified from text. We consider here automatically annotating biomedical documents with respect to relations and events belonging to the GRO.

GRO has two top-level categories of concepts, Continuant and Occurrent, where the Occurrent branch has concepts for processes that are related to the regulation of gene expression (e.g. Transcription, RegulatoryProcess), and the Continuant branch has concepts mainly for physical entities that are involved in those processes (e.g. Gene, Protein, Cell). It also defines semantic relations (e.g. hasAgent, locatedIn) that link the instances of the concepts.

The representation involves three primary categories of annotation elements: entities (i.e. the instances of Continuant concepts), events (i.e. those of Occurrent concepts) and relations. Mentions of entities in text can be either contiguous or discontinuous spans that are assigned the most specific and appropriate Continuant concepts (e.g. TranscriptionFactor, CellularComponent). Event annotation is associated with the mention of a contiguous span in text (called event trigger) that explicitly suggests the annotated event type (e.g. "controls" - RegulatoryProcess). If a participant of an event, either an entity or another event, can be explicitly identified with a specific mention in text, the participant is annotated with its role in the event. In this task, only two types of roles are considered, hasAgent and hasPatient, where an agent of an event is an entity that causes or initiates the event (e.g. a protein that causes a regulation event), and a patient of an event is an entity on which the event is carried out (e.g. the gene that is expressed in a gene expression event) (Dowty, 1991). Relation annotation is to annotate other semantic relations (e.g. locatedIn, fromSpecies) between entities and/or events, i.e. those without event triggers. An example annotation is shown in Figure 1.

The annotation of Continuant concepts has been considered for a long time and has well established methods relying on large dictionaries. GRO task has provided these annotations and only evaluates events and relations detection, including the triggers of events. We produce the annotation in two steps. The first step takes as input a biological text and the corresponding Continuant concepts and produces Occurent concepts (event triggers and their types). We provide two different formalizations of this problem: one

[2] `http://dbpedia.org/Lexicalizations`

[3] accessible on http://2013.bionlp-st.org/tasks

[4] "Event" is taken here in a biological sense, which may not fit to the state-event-process distinction or other linguistic views

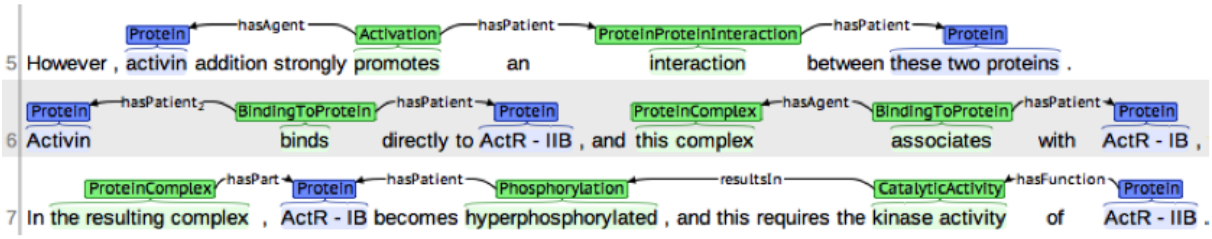

Figure 1: Example annotations from the GRO corpus (Kim et al., 2013).

as a named entity recognition problem, and the other as a machine translation problem. The second step takes as input the text and both Continuant and Occurrent concepts (predicted in step 1) and predicts relations between them. Relations are either: (a) an "event argument role" relation (hasAgent, hasPatient) between an Occurent concept and another concept, or (b) one of a small set of predefined relations between two concepts that do not involve trigger words (encodes, hasFunction, locatedIn, precedes, hasPart, resultsIn, fromSpecies, startsIn, endsIn)[5] We formalize this problem as a multi-class classification problem and solve it using a discriminative maximum-entropy classifier.

3 Step One: Event Annotation

In this step, event triggers (continuous span of text) are identified and given a label from the Occurrent concepts (98 label in total). We formalize this task as text segmentation and labeling, and compare two approaches to solve it: named-entity recognition approach and machine translation approach.

3.1 Event detection as named-entity recognition

A direct formalization of the event detection task is as named-entity recognition (hence named NER4SA). The NER task is to locate and classify elements of text into pre-defined categories. In our case, the elements are contiguous segments representing biological events, and the categories are their corresponding ontology-based occurrent labels. Conditional random fields (CRF), which represents the state of the art in sequence labeling, are widely used for NER (Finkel et al., 2005). This is mainly because they allow for discriminative training benefiting from manually annotated examples, and because of their ability to take the sequential structure into consideration through the flow of probabilistic information during inference. Here, the input sequence $\mathbf{x} = (x_1, ..., x_n)$ represents the words, and the output sequence $\mathbf{y} = (y_1, ..., y_n)$ represents the corresponding labels. The labels we use are the ontology-based Occurrent corresponding to events, combined with a segmentation marker in order to capture annotations possibly spanning multiple words. These markers are 'B' for beginning of event, 'I' for inside an event and 'O' for outside an event.

CRF is powerful in allowing for a wide range of features to be considered in the model. However, it rapidly becomes time and memory consuming when incorporating wide-range dependencies between labels. Therefore, in our experiment, we use a linear-chain CRF (bi-gram label dependency) with features including the current word as well as prefix and suffix character n-grams up to length 2. We compare two label schemes, one containing the 'B', 'I', and 'O' markers (called BIO) and a simpler 'I', and 'O' scheme (called IO).

Table 1 summarizes the results using the following settings: the training data and half of the development data from GRO task is taken to train CRF models, and the rest half development data is taken as test. We use the Stanford NER recognizer for the implementation [6]. The performance of the system varies significantly from an event trigger to another. For example, "GeneExpression" is well characterized and relatively easily detected as indicated by an F-measure of 88%, while "Disease" has a very bad recall resulting in a low F-measure of 21%. The majority of triggers such as "BindingToProtein" and "PositiveRegulation" lie in the middle. "RNASplicing" was not recognized at all, which is partially due to its

[5]Not all these relation types are present in the training and development data.

[6]http://nlp.stanford.edu/software/CRF-NER.shtml

Trigger	Precision		Recall		F-measure		TP		FP		FN	
	IO	BIO	IO	BIO	IO	BIO	IO	BIO	IO	BIO	IO	BIO
BindingToProtein	0.86		0.60		0.71		18		3		12	
Disease	0,67		0.13		0.21		2		1		14	
GeneExpression	0.85		0.92		0.88		23		4		2	
PositiveRegulation	0.79		0.61		0.69		30		8		19	
RNASplicing	0.00		0.00		0.00		0		1		4	
Localization	0.00	**0.50**	0.00	**0.13**	0.00	**0.20**	0	**1**	**1**	**1**	8	**7**
CellDeath	**1.00**	**1.00**	0.33	**0.67**	0.50	**0.80**	1	**2**	0	0	2	**1**
RegulatoryProcess	0.69	**0.75**	**0.39**	**0.39**	0.50	**0.51**	9	9	4	3	14	14
Aggregated	0,76	**0.77**	0,43	**0.44**	**0.556**	**0.563**	136	**138**	42	**41**	175	**173**

Table 1: Event detection as NER results. TP is for true positive, FP for false positive, and FN for false negative.

small number of occurrences in the data. On the aggregated class of (all) event triggers, the best result is obtained using the BIO scheme: 56.3% F-measure with a precision of 77% but with a weaker recall (44%). However, as given in the first block of Table 1, in most of the case IO and BIO schemes resulted in a comparable performance for triggers such as "BindingToProtein" and "Disease". But there are three cases (second block of Table 1) where a more fine-grained representation BIO slightly outperformed the basic IO representation. These results suggest that the segmentation scheme is of little importance for the performance of NER4SA.

3.2 Event detection as phrase-based SMT

In this section, we model the semantic annotation of specialized documents as a phrase-based statistical machine translation task (hence named SMT4SA). This modeling provides a potential advantage compared to the CRF approach due to its capacity to recognize (possibly complex) phrases as the relevant textual units to translate (annotate for our task). However, it is more difficult to incorporate arbitrary features into the model. The simple idea in SMT4SA is to consider an initial unannotated text as if it was written in a "foreign" language, and the annotated text as the target "translated" text. Formally speaking, two sentences $\langle s_1, s_2 \rangle$ are given in two languages L_1 and L_2: L_1 is English and $L_2 = L_1 \cup Voc(O)$ is the union of English and the vocabulary of the ontology $Voc(O)$ used as semantic tagset.[7] We say that s_2 is an annotated version of s_1 if it is obtained by replacing some sequences of English words in s_1 by elements of $Voc(O)$ as shown in the following Table 2.

Language L_1:	The corresponding **gene** was assigned to **chromosome** **14q31**, the same region where genetic alterations have been associated with several abnormalities of **thyroid hormone response**.
Language L_2:	The corresponding **TTGene** was assigned to **TTChromosome**, the same region where genetic alterations have been associated with several abnormalities of **TTOrganicChemical** **TEResponseProcess**.

Table 2: L1 and L2 languages (TT and TE escapes mark entities and events)

Several steps are performed in order to construct a phrase-based SMT (Koehn et al., 2003a). Word alignments are first computed from paired sentences, then phrase pairs are extracted such that no word inside the phrase pair is aligned to a word outside it; these extracted phrase pairs are stored in a phrase table with a set of features quantifying their quality. Such features include the conditional translation probability typically computed as normalized phrase frequencies in the corpus. One the system is trained, the translation process is carried out as a search for the best target sentence under a log-linear scoring function that combines several features. The scaling parameters of this function are tuned discrimina-

[7] To differentiate elements of $Voc(O)$ and the plain English vocabulary, names from O are preceded by an escape character sequence in $Voc(O)$.

tively to optimize the translation performance on a small set of paired sentences. Given a sentence to be translated, it has to be segmented into phrases which are then individually translated, and last reordered to fit the typical order of the target language. Applied to semantic annotation, the translation relation is monotonic (i.e. involves no reordering) and many elements are identical to their translation. The training data we use provides one-to-one correspondence between the words and their label which allows us to compute exact word alignments between source and target sentences. The possibility to produce good annotations when plain lexical information is ambiguous relies on the learning algorithm and the projection of its results on the text, inasmuch it takes the context into account for disambiguation. Note also that the model accounts for tokens which must not be annotated (they are learned to be identically translated). SMT systems typically incorporate a language model (LM) which helps selecting the most probable annotated sentence from the large set of possibilities, and the phrase table functions as a sophisticated dictionary between the source and target languages. We use the KenLM language model Toolkit (Heafield et al., 2013) to train a language model for our experiments. To construct the phrase table we use the relatively simple but effective method defined in (Koehn et al., 2003b) but we use exact word alignment which we compute separately. The decoding is done by a beam search as implemented by Moses (Koehn et al., 2007). To localize the precise positions of semantic annotations predicted, we use the translation alignment between the two texts provided at the word level in the output of Moses. For example, giving "15-14 16-14" in the alignment for a sentence means that the 15th and 16th words in the original are replaced by the 14th word in the translated file. If the 14th word belongs to $Voc(O)$, such as $TTGene$, the concept $Gene$ is the semantic label associated to the 15th and 16th words of the original text.

3.2.1 Evaluation

We performed several experiments in order to discover which information helps obtaining the best accuracy. The input and output languages are called respectively L1 and L2, and varying these languages is the mean to focus on different subsets of the annotations. Due to the presence of Continuant annotations (c-annotations for short) in the input, the vocabulary of both L1 and L2 is extended beyond natural language in most experiments – this is more the case for L2 than it is for L1. 'Event trigger annotation' is henceforth abbreviated as et-annotation. For evaluation, two measures are used, one less requiring than the other: a positive annotation has either the same label and the same endpoints as a reference label (exact match), or at least one of these criteria is satisfied ('AL1 match'), provided that the positions, at least, intersect. The results are summarized in Table 3. In Table 3, 'expe1' is the main experiment, working exactly in the conditions proposed by the reference task: L1 has c-annotations and L2 has both c-and et-annotations. It can be compared to the aggregated results in table 1. Some variants have been made to separate the role of different factors. In 'expe2', L1 has no annotations at all and correspond to the raw input text, and L2 has everything, i.e., c- and et- ones. The expe2-a line gives a global result of evaluating the prediction of c- and et-annotations together: F-measures is 0.16 points below 'expe1', which is an important loss. However, computing the scores separately for the two kinds of annotation in the L2 language refines the view : the c-annotations (expe2-c line) are much worse than the et-ones (expe2-b line), which have only lost .03 points with respect to 'expe1'. From this, we conclude that c-annotations in L1 (as used in 'expe1') do not help much to learn et-annotations.

Analyzing the conditions of 'expe2', it can be seen that including the c-annotations from the references in L2 provide helpful information via the inverse probabilities used as a feature in the phrase table. So we made two more experiments to check each type of annotation by itself. In 'expe3', L1 is the unannotated text and L2 has only c-annotations. A slight improvement is observed on the F-measure of AL1 relative to 'expe2-c', while the exact case gets the same score. In fact, Moses suggests 20% more annotations but the ratio of true positive is worse. In 'expe4', L1 is the text and L2 has only et-annotations. The results are 0.02 points below 'expe1' and close to 'expe2-b', which proves that knowing c-annotations does not help us much to detect events triggers in this setting (note that c-annotations are used to detect events arguments in the next section). It also clearly shows that c-annotations are much harder to learn and that dictionaries or similar lexicon-based methods are more suitable.

The following experiments, namely 'exp5' and 'exp6' have no annotations in L1 compared to 'expe4'

	#ref	#mo	#MP	#PG	#LG	#PLG	#AL1	FPL	FAL1
expe1	314	301	250	215	209	188	236	0.61	0.77
expe2-a	1229	869	734	520	594	476	638	0.45	0.61
expe2-b	313	328	248	210	214	190	234	0.59	0.73
expe2-c	916	541	468	310	391	286	415	0.39	0.57
expe3	916	647	533	334	444	310	468	0.40	0.60
expe4	313	329	253	217	213	191	239	0.60	0.74
expe5	313	242	204	175	174	158	191	0.57	0.69
expe6	313	306	246	210	204	181	233	0.58	0.75

The headers

#ref	nbr of annotations in the reference	#PLG	nbr of exact (pos- and lab-good) matches
#mo	nbr of annotations in moses output	#AL1	nbr of matches with at least one good attribute
#MP	nbr of matches (meeting pairs)	FPL	Fmesure - exact case
#PG	nbr of position-good matches	FAL1	Fmesure - at least one case
#LG	nbr of label-good matches		

Table 3: The results of experiments on event detection as phrase-based SMT.

but only et-annotations in L2. In these experiments we use factored translation models (Koehn and Hoang, 2007) as implemented in Moses. Factors allow for incorporating supplementary information, in addition to the actual words, into the model. A simple analysis suggests that being an event term could be correlated to the nature of the word (favored by being a verb) or to the kind of dependency it enters in. We therefore added part-of-speech tags and grammatical dependency labels, computed from dependency trees, to L1. In 'expe5', the three L1 factors are compared altogether to L2 while in 'expe6' they are compared independently (and successively) to 'expe6'. In the first case, the performance drops by .03 to .06 points compared to 'expe4'. The second case has small effects on the two F-measures. Finally, using factor models in our settings does not improve the recognition of event terms.

To summarize, using c-annotations in L1, c- and et-annotations in L2 provides the best result, slightly better for et-annotations alone than if c-annotations are omitted. In these settings, et-annotation reaches a precision of 62% and a recall of 59% in the exact case (78% and 75% in the approximate one). We find 60% of exact positives; nearly 40% of the obtained annotations are not exact. Among these annotations, 15% captured at least one characteristic.

The predicted annotations obtained by both NER4SA and SMT4SA are then supplied to the next step in the pipeline. This second step in which relations and event arguments are computed is discussed in the next section.

4 Step Two: Relations and Event Arguments Annotation

In the second step of the pipeline, we take the output of the first step, namely the detected events, and we predict their arguments. We also predict other relations in the text.

The essential difference between the extraction of relations and that of event arguments is that relations link exactly two locations in the text while events link a variable number of locations and are supported by triggers. Nevertheless, we use a unified representation for both events and relations. A relation is a labeled link between two elements in the text. Examples of relation labels include '*locatedIn*' and '*fromSpecies*'. An event is a set of labeled relations between the event trigger (detected in step 1 of the pipeline) to an event argument which is another element of the text. Event-to-argument relations are labeled either '*hasAgent*' or '*hasPatient*'. Therefore, the problem of relation extractions boils down to a multi-class classification problem of candidate links. A candidate link involves two c- or et-annotations and is labeled by the biological relation name in the first case, or by an event argument role when its source is an event trigger. Note that the same event trigger may have several agent or patient roles.

4.1 A multi-class classification approach

For each candidate link between two elements of the text, we predict a label among 'none' (which indicate no link), '*hasAgent*', '*hasPatient*', '*locatedIn*', etc. Although we use the same representation for both event arguments annotation and relation annotation, we use two distinct multi-class classifier. The first classifier locate the arguments of each detected event and identify their roles. Event arguments can be Continuant concepts or other events. The second classifier extracts and label relations between any two concepts which can be Continuant or events. We perform these two tasks independently and combine their predictions afterward. For event arguments annotation: for each detected event, we assign one of the labels '*hasAgent*', '*hasPatient*', '*no-relation*' to all other entities. Similarly for relation annotation: for each pair of c- or et-annotations we predict a label which is either the label of the binary relation or the special label 'no-relation'. We use an implementation of a maximum-entropy classifier called Wapiti[8] (Lavergne et al., 2010). The set of features we used contains lexical and morpho-syntactic features extracted from the pair of entities in question. This include their lexical identities as they appear in the document as well as the ontology labels assigned to them. We also include the part-of-speech tags of involved words. Additionally, we include positional features such as the distance between the words in the document, computed as the number of words separating them, as well as their relative positions indicating which word precedes the other in the text. Furthermore, we use compound features resulting from combining pairs of the individual features.

4.2 Evaluation

The reference result has much more 'No' than 'Yes', and labeling randomly while respecting the proportion would give a good score for the No. So in the evaluation the numbers of true positives, false positives and false negatives only account for 'Yes' answers. The criterion is an exact match (label and position) at each end of the link. Table 4 gives the results for the relations appearing in our test set. The number of occurrences of each relation in the reference is pointed out. Except for the sparse '*hasFunction*', the precision is at least 57% and higher for relations which have the greatest number of occurrences. For recall, however, only '*fromSpecies*' relation has an important recall.The mean precision is 80% and the mean recall is 37%, which yields a F-measure of 50%.

Relation	# of occurrences	Precision	Recall	F-measure
locatedIn	182	0.73	0.26	0.38
encodes	46	0.57	0.21	0.31
hasPart	178	0.77	0.26	0.39
fromSpecies	172	0.90	0.69	0.78
hasFunction	24	0.20	0.08	0.12

Table 4: Detection of relations

The annotation of events presents seemingly more difficulties than relations: the precision is at best 60% for a much higher number of occurrences. The recall has the same order of magnitude for the agent role, and is better for the patient role which has twice more occurrences. The mean precision is 58%, and the mean recall is 36%. In the pipeline evaluation presented in the next section, errors due to event recognition will accumulate with errors proper to relation annotation.

Class	# of occurrences	Precision	Recall	F-measure
hasPatient	562	0.61	0.43	0.50
hasAgent	258	0.46	0.20	0.28

Table 5: Detecting arguments of events

[8] `http://wapiti.limsi.fr`

5 Pipeline Evaluation

The pipeline evaluation compares the relations and events obtained at the end of the pipeline to the reference. We have implemented the algorithm defined in the task description, and applied it to one unused half of the development data. In this evaluation, the data consist in 175 documents for training (of which 25 are reserved for Moses for tuning) and 25 for testing.

	Events			**Relations**			**Both**		
Event detection	Pr	Rc	F1	Pr	Rc	F1	Pr	Rc	F1
NER4SA	.20	.10	.13	.80	.30	.44	.44	.19	.26
SMT4SA	.14	.13	.13	.80	.30	.44	.32	.21	.25
SMT4SA ∪ NER	.16	.22	.19	.80	.30	.44	.29	.26	.27

Table 6: Pipeline precision, recall and F-measure using strict matching for the NER4SA and SMT4SA approaches for event detection, and for their combination.

Relation detection has roughly the same figures as in table 4. The combination of event detection and arguments annotation obtains the same F-measure for both detection methods proposed, so the 5% point advantage of the second when tested out of the pipeline disappears here. Interestingly, using a combination (union) of the outputs of the NER and SMT approaches results in improvements in recall (and f1) over each approach in isolation.

6 Related work

Some effort has been dedicated to the recognition of ontology concepts in biomedical literature. This includes TextPresso (Muller et al., 2004) and GoPubMed (Doms and Schroeder, 2005). These approaches are based on term extraction methods to find the ontology concepts occurrences, together with some terminological variations. Systems like (Rebholz-Schuhmann et al., 2007) and FACTA (Tsuruoka et al., 2008) collect and display co-occurrences of ontology terms. However, they do not extract events and relations of the semantic types defined in ontologies. For event and relation extraction, (Klinger et al., 2011) use imperatively defined factor graphs to build Markov Networks that model inter-dependencies between mentions of events within sentences, and across sentence-boundaries. OSEE (jae Kim and Rebholz-Schuhmann, 2011) is a pattern matching system that learns language patterns for event extraction. Most similar to our work, is the TEES 2.1 system (Björne and Salakoski, 2013) which is based on multi-step SVM classifiers that learns event annotation by first locating triggers then identifying event arguments and finally selecting candidate events.

7 Conclusion

In this work, we have proposed a pipeline for annotating documents with domain sepcific ontologies and tested it on the BioNLP'13 GRO task. The two-step pipeline gives a flexible modeling choice, and is realized by different inner components. For the first step, the sequence labeling and phrase-based statistical machine translation approaches are applied. And we conducted detailed experiments to test different settings, from which we can conclude the following findings: (1) For the event recognition task, NER4SA, much computationally expensive due to its model complexity, did not result in higher scores than SMT4SA in terms of F-measure. It did give better precision, however at the expense of the recall. This shows that SMT4SA is a good practical modeling method for the task. (2) For SMT4SA, the extra features added by factored learning did not boost the system much, which means that a basic setting can capture the essential quality of the system. (3) For the relation detection based on the output of the pipeline, we obtained reasonable scores for events and relations. Interestingly, NER4SA, SMT4SA, or their combination did affect the detection of events, but not relations which is step-one independent. And the combination has had a better performance.

Acknowledgements

We are thankful to the reviewers for their comments. This work is part of the program Investissements d'Avenir, overseen by the French National Research Agency, ANR-10-LABX-0083, (Labex EFL). We acknowledge financial support by the DFG Research Unit FOR 1513, project B1.

References

[Beisswanger et al.2008] Elena Beisswanger, Vivian Lee, Jung jae Kim, Dietrich Rebholz-Schuhmann, Andrea Splendiani, Olivier Dameron, Stefan Schulz, and Udo Hahn. 2008. Gene regulation ontology (gro): Design principles and use cases. In *MIE*, volume 136 of *Studies in Health Technology and Informatics*, pages 9–14. IOS Press.

[Björne and Salakoski2013] Jari Björne and Tapio Salakoski. 2013. Tees 2.1: Automated annotation scheme learning in the bionlp 2013 shared task. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 16–25, Sofia, Bulgaria, August. Association for Computational Linguistics.

[Chiang2007] David Chiang. 2007. Hierarchical phrase-based translation. *Comput. Linguist.*, 33:201–228.

[Ciravegna et al.2004] Fabio Ciravegna, Sam Chapman, Alexiei Dingli, and Yorick Wilks. 2004. Learning to harvest information for the semantic web. In *Proceedings of ESWS'04*.

[Cucerzan2007] Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on wikipedia data. In *Proceedings of EMNLP-CoNLL'07*, pages 708–716.

[Dill et al.2003] Stephen Dill, Nadav Eiron, David Gibson, Daniel Gruhl, R. Guha, Anant Jhingran, Tapas Kanungo, Sridhar Rajagopalan, Andrew Tomkins, John A. Tomlin, and Jason Y. Zien. 2003. Semtag and seeker: bootstrapping the semantic web via automated semantic annotation. In *Proceedings of WWW '03*, pages 178–186.

[Doms and Schroeder2005] Andreas Doms and Michael Schroeder. 2005. Gopubmed: exploring pubmed with the gene ontology. *Nucleic Acids Research*, 33(Web-Server-Issue):783–786.

[Dowty1991] David Dowty. 1991. Thematic proto-roles and argument selection. *Language*, 67:547–619.

[Finkel et al.2005] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of ACL'05*, pages 363–370.

[Guha and McCool2003] R Guha and R McCool. 2003. Tap: A semantic web test-bed. *Web Semantics Science Services and Agents on the World Wide Web*, 1(1):81–87.

[Heafield et al.2013] Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 690–696, Sofia, Bulgaria, August.

[jae Kim and Rebholz-Schuhmann2011] Jung jae Kim and Dietrich Rebholz-Schuhmann. 2011. Improving the extraction of complex regulatory events from scientific text by using ontology-based inference. *J. Biomedical Semantics*, 2(S-5):S3.

[Kim et al.2013] Jung-Jae Kim, Xu Han, Vivian Lee, and Dietrich Rebholz-Schuhmann. 2013. Gro task: Populating the gene regulation ontology with events and relations. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 50–57, Sofia, Bulgaria, August. Association for Computational Linguistics.

[Kiryakov et al.2004] Atanas Kiryakov, Borislav Popov, Ivan Terziev, Dimitar Manov, and Damyan Ognyanoff. 2004. Semantic annotation, indexing, and retrieval. *Journal of Web Semantics*, 2:49–79.

[Klinger et al.2011] Roman Klinger, Sebastian Riedel, and Andrew McCallum. 2011. Inter-event dependencies support event extraction from biomedical literature. Mining Complex Entities from Network and Biomedical Data (MIND), European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD).

[Koehn and Hoang2007] Philipp Koehn and Hieu Hoang. 2007. Factored translation models. In *EMNLP-CoNLL*, pages 868–876. ACL.

[Koehn et al.2003a] Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003a. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 48–54, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Koehn et al.2003b] Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003b. Statistical phrase-based translation. In *HLT-NAACL*, pages 127–133.

[Koehn et al.2007] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of ACL'07*, pages 177–180.

[Lavergne et al.2010] Thomas Lavergne, Olivier Cappé, and François Yvon. 2010. Practical very large scale CRFs. In *Proceedings the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 504–513. Association for Computational Linguistics, July.

[Liu et al.2011] Xiaohua Liu, Shaodian Zhang, Furu Wei, and Ming Zhou. 2011. Recognizing named entities in tweets. In *Proceedings of HLT '11*, pages 359–367.

[LiveMemories2010] LiveMemories. 2010. Livememories: Second year scientific report. Technical report, LiveMemories, December.

[Marcu and Wong2002] Daniel Marcu and William Wong. 2002. A phrase-based, joint probability model for statistical machine translation. In *Proceedings of EMNLP'02*, pages 133–139.

[Mendes et al.2011] Pablo N. Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. 2011. DBpedia Spotlight: Shedding light on the web of documents. In *Proceedings of I-Semantics'11*.

[Mihalcea and Csomai2007] Rada Mihalcea and Andras Csomai. 2007. Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of CIKM'07*, pages 233–242.

[Muller et al.2004] H. Muller, E. Kenny, and P. Sternberg. 2004. Textpresso: An ontology-based information retrieval and extraction system for biological literature. *PLoS Biology*, 2(11):1984–1998.

[Nazarenko et al.2011] Adeline Nazarenko, Abdoulaye Guissé, Franois Lévy, Nouha Omrane, and Sylvie Szulman. 2011. Integrating written policies in business rule management systems. In *Proceedings of RuleML'11*.

[Och and Ney2003] Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, pages 19–51.

[Rebholz-Schuhmann et al.2007] Dietrich Rebholz-Schuhmann, Harald Kirsch, Miguel Arregui, Sylvain Gaudan, Mark Riethoven, and Peter Stoehr. 2007. Ebimed - text crunching to gather facts for proteins from medline. *Bioinformatics*, 23(2):237–244.

[Stolcke2002] Andreas Stolcke. 2002. Srilm — an extensible language modeling toolkit. In *In Proceedings of ICSLP'02*, pages 901–904.

[Tsuruoka et al.2008] Y Tsuruoka, J Tsujii, and S Ananiadou. 2008. Facta: a text search engine for finding associated biomedical concepts. *Bioinformatics*, 24(21):2559–2560, November.

[Uren et al.2006] Victoria S. Uren, Philipp Cimiano, José Iria, Siegfried Handschuh, Maria Vargas-Vera, Enrico Motta, and Fabio Ciravegna. 2006. Semantic annotation for knowledge management: Requirements and a survey of the state of the art. *J. Web Sem.*, 4(1):14–28.

[Wang2009] Yefeng Wang. 2009. Annotating and recognising named entities in clinical notes. In *ACL/AFNLP (Student Workshop)*, pages 18–26.

[Welty and Ide1999] Christopher Welty and Nancy Ide. 1999. Using the right tools: Enhancing retrieval from marked-up documents. In *Journal Computers and the Humanities*, pages 33–10.