Proceedings of the 15th Joint ACL - ISO Workshop on Interoperable Semantic Annotation (ISA-15)

May 23, 2019

at IWCS 2019
Gothenburg, Sweden

Harry Bunt, editor

Proceedings of the Fifteenth Joint ACL - ISO Workshop on
Interoperable Semantic Annotation (ISA-15)

Workshop at IWCS 2019
Gothenburg, Sweden, May 23, 2019

TiCC, Tilburg center for Cognition and Communication
Tilburg University, The Netherlands

ISBN 9789074029384

Table of Contents

Organizing Committee

Harry Bunt (chair)	Tilburg University
Nancy Ide	Vassar College, Poughkeepsie, NY
Kiyong Lee	Korea University, Seoul
James Pustejovsky	Brandeis University, Waltham, MA
Laurent Romary	INRIA/Humboldt Universität Berlin

Programme Committee

Jan Alexandersson	DFKI, Saarbrücken
Ron Artstein	University of Southern California
Harry Bunt (chair)	Tilburg University
Nicoletta Calzolari	ILC-CNR, Pisa
Jae-Woong Choe	University of Korea, Seoul
Robin Cooper	University of Gothenburg
Thierry Declerck	DFKI, Saarbrücken
David DeVault	University of Southern California
Jens Edlund	KTH, Stockholm University
Alex Chengyu Fang	City University Hong Kong
Robert Gaizauskas	University of Sheffield
Kallirroi Georgila	University of Southern California
Jan Hajic	Charles University, Prague
Koiti Hasida	Tokyo University
Nancy Ide	Vassar College
Elisabetta Jezek	Università degli Studi di Pavia
Simon Keizer	Vrije Universiteit Brussel
Kiyong Lee	University of Korea, Seoul
Paul Mc Kevitt	University of Ulster
Adam Meyers	New York University
Philippe Muller	IRIT, Université Paul Sabatier, Toulouse
Patrizia Paggio	L-Università tà Malta
Volha Petukhova	Universität des Saarlandes, Saarbrücken
Massimo Poesio	Queen Mary University, London
Rashmi Prasad	University of Wisconsin, Milwaukee
Laurent Prévot	Aix-Marseille University
James Pustejovsky	Brandeis University
Ines Rehbein	Leibniz Science Campus, Universität Heidelberg
Laurent Romary	INRIA/Humboldt Universität Berlin
Ielka van der Sluis	University of Groningen
Manfred Stede	Universität Potsdam
Matthew Stone	Rutgers, the State University of New Jersey
Thorsten Trippel	University of Bielefeld
Carl Vogel	Trinity College Dublin
Piek Vossen	Vrije Universiteit Amsterdam
Menno van Zaanen	Tilburg University
Annie Zaenen	Stanford University
Heike Zinsmeister	Universität Hamburg
Sandrine Zufferey	Université de Berne

Workshop Program

08.45 -- 09:00 Registration

09:00 – 09:10 Opening by the Program Committee Chair

09:10 -- 09:40 Giuseppe Abrami, Alexander Mehler, Andy Lücking, Elias Rieb and Philipp Helfrich: *TextAnnotator: A flexible framework for semantic annotation*
09:40 -- 10:10 Robert Gaizauskas and Tarfah Alrashid: *SceneML: A Proposal for Annotating Scenes in Narrative Text*
10:10 – 10:30 Tarfah Alrashid and Robert Gaizauskas: *Annotating and Recognising Visually Descriptive Language*

10:30 – 11:00 Coffee break

11:00 -- 11:30 Harry Bunt: *Plug-ins for content annotation of dialogue acts*
11:30 – 12:00 James Pustejovsky, Kiyong Lee and Harry Bunt: *The Semantics of ISO-Space*
12:00 -- 12:20 Elisabetta Jezek , Edoardo Maria Ponti and Bernardo Magnini: *Evaluating distributional representations of verb semantic selection*

12:20 – 13:50 Lunch break

13:50 -- 14:20 Jaipal Goud , Pranav Goel, Alok Debnath, Suhan Prabhu and Manish Shrivastava: *A Semantico-Syntactic Approach to Event-Mention Detection and Extraction in Hindi*
14:20 – 14:40 Jaipal Goud, Pranav Goel, Allen Antony and Manish Shrivastava: *Leveraging Multi Lingual Resources for Open-Domain Event Detection*
14:40 -- 15:00 Merle Pfau and Heike Zinsmeister *Annotating semantic frames on top of syntactic dependencies*
15:00 -- 15:30 Shanshan Liu, Haitao Wang, Xinyu Cao, Kiyong Lee and Tianyong Hao: *The Semantic Annotation of Measurable Quantitative Information*

15:30 Closing

15:30 Tea break

TEXTANNOTATOR: A flexible framework for semantic annotations

Giuseppe Abrami
Goethe University Frankfurt
abrami@em.uni-frankfurt.de

Alexander Mehler
Goethe University Frankfurt
mehler@em.uni-frankfurt.de

Andy Lücking
Goethe University Frankfurt
luecking@em.uni-frankfurt.de

Elias Rieb
Goethe University Frankfurt
s0294036@stud.uni-frankfurt.de

Philipp Helfrich
Goethe University Frankfurt
helfrich@cs.uni-frankfurt.de

Abstract

Modern annotation tools should meet at least the following general requirements: they can handle diverse data and annotation levels within one tool, and they support the annotation process with automatic (pre-)processing outcomes as much as possible. We developed a framework that meets these general requirements and that enables versatile and browser-based annotations of texts, namely TEXTANNOTATOR. It combines NLP methods of pre-processing with methods of flexible post-processing. In fact, machine learning (ML) requires a lot of training and test data, but is usually far from achieving perfect results. Producing high-level annotations for ML and post-correcting its results are therefore necessary. This is the purpose of TEXTANNOTATOR, which is entirely implemented in ExtJS and provides a range of interactive visualizations of annotations. In addition, it allows for an flexible integration of knowledge resources. The paper describes TEXTANNOTATOR's architecture together with three use cases: annotating temporal structures, argument structures and named entity linking.

1 Introduction

Among other things, annotation tools must be able to take into account the diversity of data and its multi-level annotation, be easy to use and enable the convertibility of data formats and files (Dipper et al., 2004, p. 55 f.). In view of the current achievements of automatic annotations as a result of *Natural Language Processing* (NLP), one can add to this list of requirements the integration of NLP-based pre- and post-processing to support and facilitate human annotation. To avoid using multiple tools when addressing different annotation tasks and to combine efficient pre-processing with flexible post-annotation we developed TEXTANNOTATOR as a framework that enables versatile and browser-based annotations of texts while using the pre-processing methods of TEXTIMAGER (Hemati et al., 2016). In this paper, we compare TEXTANNOTATOR with related tools in Sec. 2, describe its architecture in Sec. 3 and its components in Sec. 4. To this end, we consider three annotation objects: temporal structure, argumentation structure and named entity linking. Finally, in Sec. 5 we conclude and in Sec. 6 we give an outlook on upcoming development steps. In order to prepare the subsequent sections, we start by outlining TEXTANNOTATOR's main features.

As a matter of fact, machine learning-based NLP tools require a lot of training and test data, but are usually far from achieving perfect results (Pinto et al., 2016; Gleim et al., 2019). Since tools propagate their errors to subsequent processing steps, TEXTANNOTATOR provides an interface for editing such errors. To this end, TEXTANNOTATOR addresses the following requirements:

a) **Customizable annotation scheme:** TEXTANNOTATOR's annotation scheme is implemented by means of native UIMA TYPE SYSTEM DESCRIPTORS (Götz and Suhre, 2004). In this way, its annotation scheme can be flexibly extended or modified, for example, to reflect changing requirements in the course of an annotation project. Such an annotation scheme has already been created by Helfrich et al. (2018) to map, for example, data structures on the level of hypergraphs.

b) **Resource management:** For managing different corpora, TEXTANNOTATOR uses the so-called *ResourceManager* (Gleim et al., 2012) which processes text files of various formats (HTML, TEI, XMI, etc.).

c) **User management:** For managing annotation projects, TEXTANNOTATOR uses the *Authority-Manager* (Gleim et al., 2012) which allows for specifying user- and group-related access permissions.

d) **Pre-processing:** In order to profit from automatic pre-processing of text corpora, TEXTANNOTATOR uses the range of tools of TEXTIMAGER (Hemati et al., 2016). This includes *inter alia* the following annotation tasks: tokenization, sentence splitting, POS tagging, lemmatization, morphological tagging, named entity recognition (NER), time recognition, sentiment analysis, disambiguation, and topic labeling. These tasks are performed for a wide range of languages using several NLP pipelines.

e) **Annotation interface:** TEXTANNOTATOR's interface offers means for selecting, annotating and visualizing information objects (text spans and their annotations). It is developed as a browser-based system using the ExtJS[1] framework. The interface has already been evaluated in Helfrich et al. (2018) by example of annotating rhetorical structures according to RST (Mann and Thompson, 1988) – see also Sec. 4.6.

f) **Storage solutions:** Managing different storage and export formats is essential for processing annotations in NLP. By means of UIMA *Type System Descriptors*, TEXTANNOTATOR uses a standard for representing such annotations (Biemann et al., 2017). As a result, annotations can be exported or saved with the help of UIMA's exchange format *XMI*. However, a purely document-based approach leads to problems with regard to redundancy management and the lack of DB-related query options: The reason being that for each document, information units have to be annotated separately and duplicated over individual annotation documents. This in turn leads to an increase in document size, which delays any queries on these documents. In order to prevent this, TEXTANNOTATOR's management of UIMA documents is based on the UIMA DATABASE INTERFACE (Abrami and Mehler, 2018). The latter provides an interface for generic database support that are not offered for UIMA documents natively, so that annotation objects can be centrally stored and referred to from individual documents.

g) **Knowledge bases:** Enriching texts with information from various knowledge source such as Wikidata (`https://www.wikidata.org/wiki/Wikidata:Main_Page`) or WordNet (`https://wordnet.princeton.edu/`, Fellbaum (1998)) is essential for many NLP tasks (e.g., NER or topic labeling (Uslu et al., 2019)). TEXTANNOTATOR allows for annotating texts with such information. This includes a wide range of knowledge resources that provide a Web API as Wikipedia, Wikidata, Wiktionary and GeoNames.

h) **Distributed and augmented annotation:** In order to enable distributed annotations in the sense that various annotators have (simultaneous) access to the same annotation files and single annotators have access to their annotation files independent from their location, the use of browser-based annotation tools has become a *de facto* standard (see also the tools covered in Sec. 2, as well as in Biemann et al., 2017). At the same time, more and more advanced devices such as smartphones, tablets, or virtual reality (VR) glasses not only support mobile and ubiquitous access, but also

[1] `https://www.sencha.com/products/extjs/`

three-dimensional and augmented reality (AR) visualizations. Anticipating future applications which increase operability due to the immersive impact of VR and AR techniques, the backend of TEXTANNOTATOR is designed to connect to platforms like VANNOTATOR (Spiekermann et al., 2018) or RESOURCES2CITY (Kett et al., 2018). This allows access to the same data in different virtual ways. Note that VANNOTATOR also meets the requirement of outdoor annotation (cf. Wither et al., 2009). Both tools, VANNOTATOR and RESOURCES2CITY, were developed to visualize and annotate multimodal information units in virtual environments, in the case of VANNOTATOR this also includes extended, three-dimensional environments.

i) **Simultaneous collaboration:** With regard to cooperative annotation tasks, the processes involved need to be coordinated. For instance, possibly complex annotation tasks have to be carried out by several annotators. Furthermore, joint annotation gives rise to a displaying challenge: annotations from other annotators as well as automatic annotations from pre-processing steps needs to be displayed. Moreover, each annotation must be fingerprinted to determine who authored it (e.g., a human annotator or a pre-processor). Since TEXTANNOTATOR's backend communicates via WebSockets, annotations can be performed in real time and platform-independently, and visualized to all participants.

Note that some of these features still undergo further extensions: the knowledge bases from g) are continuously augmented by additional ontologies, distributed and augmented annotation from h) is adapted to more tools of TEXTANNOTATOR's backend. All these further developments of TEXTANNOTATOR are available via the website `www.textannotator.texttechnologylab.org`. For each annotation component described in this paper in Sec. 4, completely annotated sample documents are available.

As explained in the following section, there exists already a number of different annotation tools for different annotation tasks. However, it is usually time-consuming or difficult to change annotation tools within a running project in order to continue working on the same document (e.g. due to a change or extension of the annotation focus). To facilitate this, we are pursuing the idea of making UIMA documents interchangeable within our uniform annotation system, so that TEXTANNOTATOR can load and process already annotated documents. In addition, any annotation results can be exported and reused by our UIMA-based tool to manage a specific project.

2 Related Work

There exists a wide range of web-based annotation tools for many annotation tasks. Annotation tools like GATE (Cunningham et al., 2013), which are not primarily web-based, are not considered here in detail (see, e.g., Wilcock, 2017 for a respective overview). In any event, because of not being based on the UIMA framework, GATE falls short of accounting for features a), partially d), and g)–i) of Sec. 1. In contrast to this, BRAT (Stenetorp et al., 2012), which has been the source of the WEBANNO annotation framework (de Castilho et al., 2014), is based on the UIMA architecture. However, BRAT and WEBANNO are specialized tools in the sense that their annotation schemes and graphical interfaces are intrinsically related to a certain class of annotation tasks. They aim at mapping continuous text spans to labels and linking them as instances of certain relations (e.g. co-reference). That is, in order to add, for instance, annotations of rhetorical discourse structure one has to change to another, presumably likewise specialized, tool. The difference between TEXTANNOTATOR and these tools is its strict implementation as a web-based solution and its modular architecture (Sec. 4) for integrating modules for special annotation tasks. That is, TEXTANNOTATOR follows a more modular approach so that extensions can be developed within the TEXTANNOTATOR framework.

At the same time, the flexible maintenance of user rights and the ability to integrate web-based knowledge resources such as Wikidata, WordNet or Wiktionary are desirable features of current annotation tools. There are tools that focus on utilizing knowledge resources in the latter sense: For instance, BABELFY (Moro et al., 2014) integrates different knowledge databases, but does not allow for post-processing of its annotations. With INCEpTION (Klie et al., 2018) the authors provide a tool for interactive and semantic annotation of texts supported by knowledge databases, providing an active learning unit supporting the annotators.

TEXTANNOTATOR is developed in such a way that it adheres to state-of-art technologies and meets real world annotation requirements (see Sec. 1), but at the same time tries to avoid architectural or conceptual limitations as known from many years of annotation experience (Gries and Berez, 2017).

3 Architecture

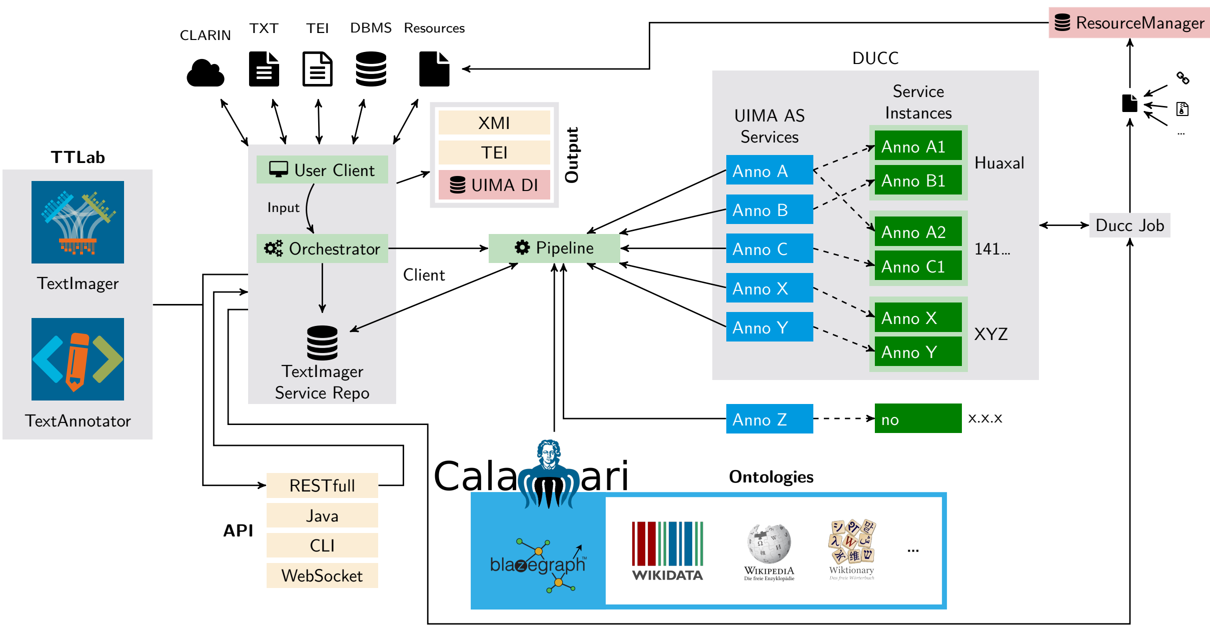

Figure 1: The system architecture with all system components used by TEXTANNOTATOR.

The architecture of TEXTANNOTATOR is primarily based on the consistent use of the UIMA framework, which supports current requirements of annotation processes (Wilcock, 2017). This means that TEXTANNOTATOR allows the pre-processing and analysis of plain text and of text corresponding to the UIMA format. Generally speaking, we distinguish two types of architectures: the architecture of underlying system components and the architecture of TEXTANNOTATOR itself. Both are described in the following sections.

3.1 System Components

The distributed system, the basis of TEXTANNOTATOR, is illustrated in Fig. 1. On the left-hand side, the two independent but interlinked main tools are shown, namely, TEXTIMAGER and TEXTANNOTATOR. Both of them access the TEXTIMAGER SERVICE REPOSITORY using a browser interface implemented in ExtJS via a RESTful webservice. In addition, the API of TEXTIMAGER can be used independently via RESTful. Every request is taken on by the so-called ORCHESTRATOR, which assembles the underlying UIMA pipelines and orders them in time (synchronous/asynchronous; hence its name). As a result of such a pre-processing pipeline, a UIMA document is created, which can be exported in different formats (XMI, TEI) as shown in Fig. 1, “Output”. Beyond that, the results can be stored by means of the UIMA DATABASE INTERFACE. However, TEXTANNOTATOR is not limited to UIMA, since TEXTIMAGER can process and convert different formats, implementing the convertibility of data formats, a requirement identified in Sec. 1.

Finally, different NLP resources can be transferred from the local system to UIMA pipelines using the web interface or the RESOURCEMANAGER (top right of Fig. 1). The gray box displays the *Apache DUCC*[2] architecture, a Linux cluster controller that allows for scaling any UIMA pipeline to high data throughput and for low latency real-time applications running on distributed systems in multiple threads. This means that large corpora can be processed much faster with DUCC than with conventional methods. This approach allows a flexible, synchronous and asynchronous pre-processing of texts, realizing the corresponding feature d) of Sec. 1.

The lower part of Fig. 1 shows the CALAMARI system, our currently developed database solution based on Blazegraph[3] for integrating various knowledge databases such as Wikidata or Wiktionary. CALAMARI aims at flexibly handling knowledge resources to meet Requirement g) of Sec. 1.

3.2 TEXTANNOTATOR

TEXTANNOTATOR consists of two main components: a component for selecting input documents together with pipelines from TEXTIMAGER to be used for pre-processing, and a component that includes all annotation modules currently provided by TEXTANNOTATOR. The former component implements three ways of selecting input documents: one can insert a plain text, load a document with the help of RESOURCEMANAGER or insert a URL to extract the corresponding web content. The second component is responsible for the annotation and visualization of input documents; both components are further described in Sec. 4. TEXTANNOTATOR's frontend is built with the help of ExtJS[4]. It makes use of a combination of the *Model View Controller* pattern and a loose data binding provided by the framework. Through a strict modularization, as forced by this pattern, new annotation and visualization components can be easily integrated into TEXTANNOTATOR. Each annotation tool is implemented in its own panel, which inherits from an underlying base panel. The latter is controlled by a controller and serves as an abstraction layer to handle events and actions that are common to all annotation tools. This allows TEXTANNOTATOR for using different visualization libraries and annotation tools while handling them on an equal footing. To this end, the annotation panels use the visualization library d3.js. New modules can be derived from given panels extending the aforementioned base panel and automatically provide an SVG after rendering. Because annotation requirements increase with the capabilities of web browsers, additional classes for new panel types such as Canvas[5], OpenGL, Unity3D[6] etc. can and will be included in future versions of TEXTANNOTATOR. All annotation panels can perform annotations of NLP documents with graphical assistance, where changes are directly realized in the underlying XMI representation of the document. These changes are also stored in the underlying UIMA database (Abrami and Mehler, 2018).

All annotations are defined as a *UIMA TypeSystemDescriptor*, which enables their porting. For all annotation components for which there was currently no *UIMA TypeSystemDescriptor available*, such descriptors were created accordingly. This concerns the annotation components TIMEANNOTATOR, ARGUMENTANNOTATOR and KNOWLEDGEBASELINKER.

In addition, a separate annotation panel for each document is created and displayed within a split window. This allows different documents to be compared with each other. Furthermore, TEXTANNOTATOR basically allows the distributed, simultaneous annotation of the same text by different users. These annotations from different users are stored in shared or separate logical user views. During post-processing, the different user annotations can be compared with each other (e.g. regarding agreement). Within the TEXTANNOTATOR this feature does not yet exist and is planned for future developments, however. This feature mainly addresses requirement i) and allows multiple users to simultaneously work on a document and mutually view changes in real time.

[2] https://uima.apache.org/doc-uimaducc-whatitam.html
[3] https://www.blazegraph.com/
[4] https://www.sencha.com/products/extjs/
[5] https://www.w3.org/TR/2dcontext/
[6] https://unity3d.com

Figure 2: Visualization of a *split window* using QUICKANNOTATOR to visualize two documents. The quick annotations can be pre-selected in the *OptionPanel* and afterwards annotated in the text. The different named entities and common nouns are distinguished by coloured highlighting. The texts shown are extracted from the *BIOfid* project on biodiversity (`https://www.biofid.de/en/`).

4 Components

TEXTANNOTATOR currently contains six annotation components: QUICKANNOTATOR, PROPOSITIONANNOTATOR, TREEANNOTATOR, TIMEANNOTATOR, ARGUMENTANNOTATOR and KNOWLEDGEBASELINKER. In order to shorten the time required to get used to the different annotation views and to create a clear user interface, each of these annotation modules uses uniform navigation and selection mechanisms as shown in Fig. 4.

As this paper is limited in size, only four components can be presented; TREEANNOTATOR has already been described in detail in (Helfrich et al., 2018). We skip PROPOSITIONANNOTATOR, that allows for the annotation of predicate-argument structures.

4.1 QUICKANNOTATOR

The so called QUICKANNOTATOR enables the fast annotation of lexical semantic structures. These currently include the annotation of named entities, common nouns and taxa as well as the annotation of sentence boundaries. This can be done by pre-selecting the types defined in the *OptionPanel* or by directly selecting the tokens. In the first case the selected token is annotated directly, in the latter case all annotation possibilities are displayed to perform the annotation. Named entities can be assigned to kinds, which are distinguished by color on display – cf. Fig. 2. Note that named entities can be annotated recursively. In this case, a "hierarchical merge" is applied and visualized by means of blue borders, as shown in Fig. 3. Furthermore, QUICKANNOTATOR can be used to create comments for the text units, which are visualized by note symbols on each token.

The primary focus of QUICKANNOTATOR is its capacity to perform annotations rapidly and easily. For more complex annotations, the targeted annotation views are designed.

4.2 TREEANNOTATOR

TREEANNOTATOR (Helfrich et al., 2018) is a graphical tool for annotating tree-like structures, in particular structures that jointly map dependency relations and inclusion hierarchies as used by RST (Mann and Thompson, 1988). Its inclusion expands the area of text data structures covered by TEXTANNOTATOR by a number of functions that go beyond sequence labeling and linking, as currently addressed

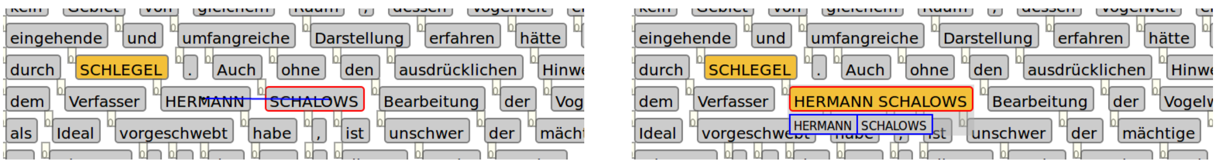

Figure 3: Creating multitokens: the left side shows two tokens manifesting the two parts of a compound proper name. The tokens are not individually annotated as *person* but combined to a multitoken which is then annotated as *person* (right). The right side also shows the combined tokens; the multitoken can be separated again at any time.

by tools such as BRAT or WEBANNO. For more details on this annotation module see Helfrich et al. (2018).

4.3 ARGUMENTANNOTATOR

ARGUMENTANNOTATOR allows the annotation of complex argumentation structures of texts, following the approach of Freeman (1991) which provides different *composition schemata*. As shown in Fig. 4, the individual arguments from the TEXTPANEL can be selected in the annotation window in order to transform them into an argumentation structure using the *schemata* from the OPTIONPANEL. The combination of these schemes is illustrated in Stede (2007, p. 113f), where the argument structure of the underlying sample text is explicated.

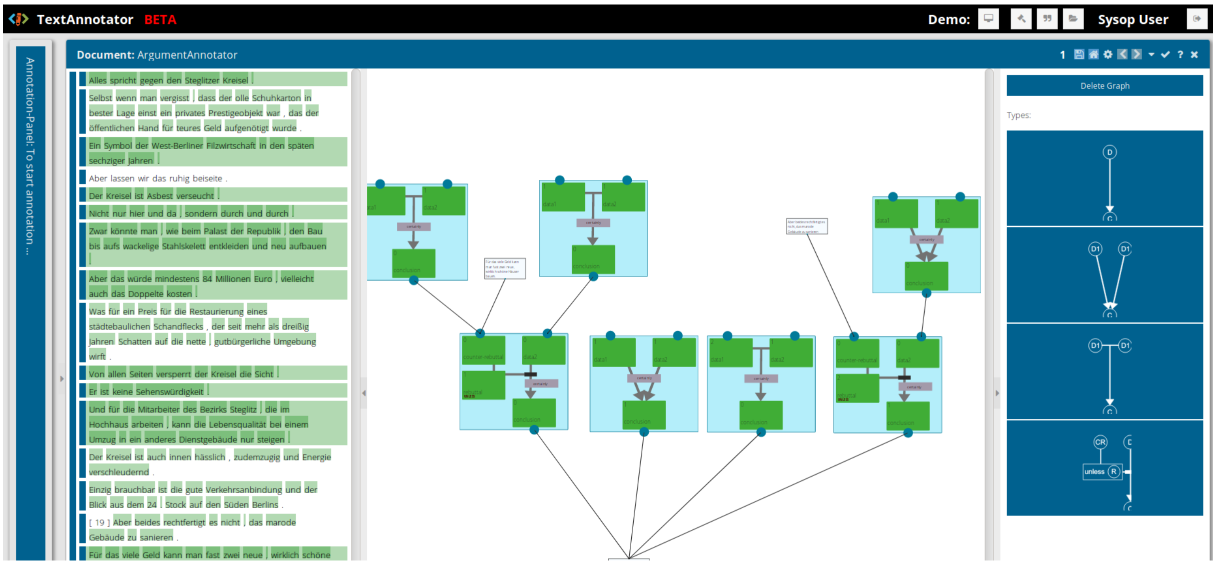

Figure 4: Annotation of argument structures. The central area (*AnnotationPanel*) of each annotation view contains the visualization of the annotations. The left area (TEXTPANEL) visualizes the raw text and highlights its paragraph, sentence and token level in order to facilitate the selection of the text segments to be annotated. The right area (OPTIONPANEL) shows the argument structures according to Freeman (1991) for annotation.

4.4 TIMEANNOTATOR

TIMEANNOTATOR (see Fig. 5) enables the annotation of basic temporal structures in texts. The underlying annotation scheme used is that of Mani and Pustejovsky (2004), according to which temporal structure is expressed in terms of the relative sequence of the events described in a text, where this relative sequence is represented as a tree. In TIMEANNOTATOR, this approach is complemented by that of

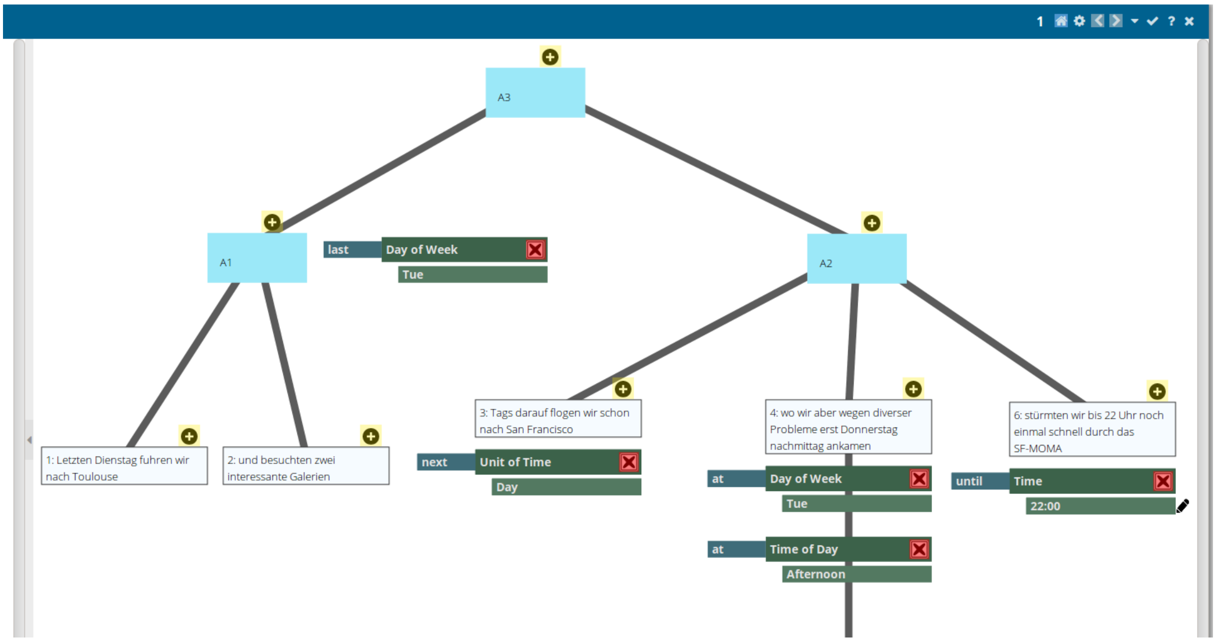

Figure 5: Annotation of temporal structures by means of TIMEANNOTATOR. Selected text segments (white) are grouped hierarchically (blue) and can be connected to temporal instances (green). Note that the *TextPanel* is collapsed in this screenshot.

Stede (2007), which also captures single points in time. Thus, each node in a time tree can receive temporal annotations in the form of concrete time stamps, time spans or relational expressions (cf. Table 1). This assignment takes place in two stages where a relative time specification is followed by a temporal instance. Subsequently, values can be selected (e.g. *Day Of Week: Mon, Thu, ...*) or entered freely (*Time, Other*). Moreover, temporal instances are not limited to text segments, but also grouped nodes can be annotated accordingly.

4.5 KNOWLEDGEBASELINKER

TEXTANNOTATOR provides an annotation panel for *Named Entity* annotation, namely KNOWLEDGE-BASELINKER (KBL). The visualization of the KBL panel was inspired by BABELFY[7]; in comparison to BABELFY, however, KBL supports the integration and linking of any number of knowledge databases.

Designed to expand on existing tools, TEXTANNOTATOR's KBL approach combines already implemented NLP tools from TEXTIMAGER with an easy to use graphical interface. Each token is automatically represented by an annotation box, which holds references and quick overviews for each linked ontology, including images, hyperlinks and short descriptions (see Fig. 6). Currently, Wikidata, Wikipedia, Wiktionary, Geonames[8], the German National Library[9], GermaNet and Babelfy (Moro et al., 2014) are accessible and can quickly be searched within TEXTANNOTATOR. For a more fine-grained NE categorization, we developed the TTLAB NAMED ENTITY TYPE system, which distinguishes 15 different NE types and 90 subtypes (see Nagel (2008); Debus (2012); Kamianets (2000); Brendler (2004); Vasil'eva (2011), Wiktionary, Urban Dictionary[10] and the ICOS List of Key Onomastic Terms[11]). In addition, relations between Wikidata elements (direct connections based on Wikidata properties) are visualized as directed edges at the bottom of the display, allowing users to quickly detect relations between annotated objects, where the type of relation is depicted as an edge label (see Fig. 6).

[7] http://babelfy.org
[8] http://www.geonames.org/
[9] http://www.dnb.de
[10] www.urbandictionary.com, accessed February 28, 2019
[11] https://tinyurl.com/y4ubpzdc, accessed February 28, 2019

Relation		Temporal instance
at		Day of Week
next		Time of Day
previous		Month
before		Day
after	×	Year
during		Time
first		Date
last		Unit of Time
		Other

Table 1: Selection options for temporal annotation in TIMEANNOTATOR: possible combinations of time relations and temporal instances.

KBL makes it possible to quickly annotate text streams by mapping proper names and common nouns to elements of ontological resources, thereby generating data that can be used to train taggers to automatically perform these tasks. This corresponds exactly to a current application scenario of KBL in the BIOfid project.

4.6 Usability

Since TEXTANNOTATOR allows for various kinds of annotations, comparing its usability with other, usually more specialized annotation tools has to focus on specific annotation tasks which define the intersection between the tools to be evaluated. So far, the usability of TEXTANNOTATOR has been tested in a comparative evaluation study regarding the annotation of rhetorical structures using TREEANNOTATOR, which is reported in (Helfrich et al., 2018). In sum, RST annotation based on TREEANNOTATOR requires less clicks, proceeds faster, and produces less errors than the corresponding annotation based on the RSTTOOL (O'Donnell, 1997, 2000). This evaluation shows that the user interface of TEXTANNOTATOR – at least in the example of TREEANNOTATOR – does justice to one of the most important features from the user's point of view, namely ease of use (Dipper et al., 2004, cf. also Sec. 1).

5 Conclusion

In this paper, we presented TEXTANNOTATOR as a novel text annotation tool that addresses different linguistic annotation tasks within the same framework. Given its user-friendliness, state-of-the-art technology, linkage of multilevel annotations, collaborative usability and connection to external knowledge resource, among others, we feel save to claim that TEXTANNOTATOR is the most advanced annotation tool currently around. However, even at this stage there are still many ideas for further improvements, some of which will be concludingly highlighted.

6 Future Work

TEXTANNOTATOR will be further developed regarding various aspects: first of all, the feature to select and annotate discontinuous text segments will be extended. Further, CALAMARI will be expanded through integrating more knowledge resources, such as domain-specific ontologies, for example, from the field of biodiversity. For all (new) visualizations provided by TEXTANNOTATOR, a LaTeX (TiKZ) export will be provided to obtain customizable LaTeX source files as well as high-quality vector graphics, thereby following the example of TREEANNOTATOR. Furthermore, a generic active learning component (e.g. Fang et al., 2017) will be implemented in TEXTANNOTATOR that supports the annotators. In addition, possibilities for collaborative annotation will be expanded. An important issue in this respect is the

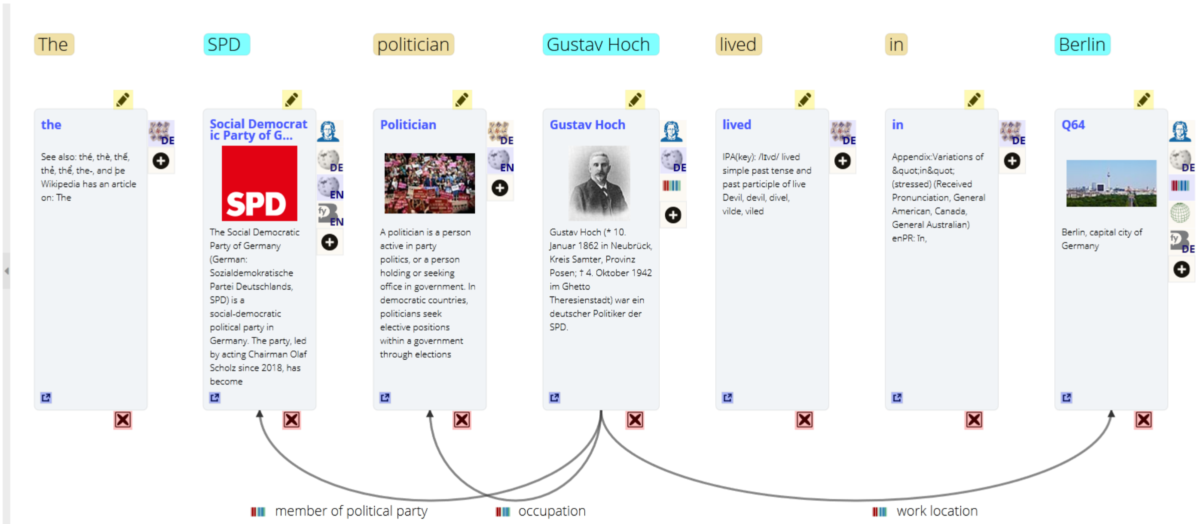

Figure 6: KNOWLEDGEBASELINKER: The selected text "The SPD politician Gustav Hoch lived in Berlin." is graphically displayed in the *AnnotationPanel*. For each recognized token from pre-processing, a query is made in the knowledge databases (in the future this will be done by CALAMARI). Under each token, the notes from the knowledge databases are visualized in groups (gray box), which can be added, edited or removed. Additionally, all tokens, which are connected by a relation in Wikidata, are visualized with a directed edge and the corresponding relation name is displayed.

easy inclusion of annotation schemes: currently, new annotation schemes can only be applied by creating new *UIMA TypeSystemDescriptor*s, which requires a restart of the underlying database. To avoid this and to ensure greater flexibility, it should be possible within TEXTANNOTATOR to utilize schemes based on single or groups of users without having to define them as individual *UIMA TypeSystemDescriptor*s. Finally, once TEXTANNOTATOR has left its current construction phase, it will be published via GitHub.

Acknowledgement

The support by the *German Research Foundation* (DFG) through the BIOfid project (`https://www.biofid.de/en/`) and by the *Federal Ministry of Education and Research* (BMBF) via the project CEDIFOR (https://www.cedifor.de/) are both gratefully acknowledged.

References

Abrami, G. and A. Mehler (2018). A UIMA Database Interface for Managing NLP-related Text Annotations. In *Proc. of LREC 2018*, Miyazaki, Japan.

Biemann, C., K. Bontcheva, R. E. de Castilho, I. Gurevych, and S. M. Yimam (2017). Collaborative web-based tools for multi-layer text annotation. In *Handbook of Linguistic Annotation*, pp. 229–256. Springer.

Brendler, S. (2004). Klassifikation der Namen. In A. Brendler and S. Brendler (Eds.), *Namenarten und ihre Erforschung. Ein Lehrbuch für das Studium der Onomastik*, Chapter 2, pp. 69–92. Hamburg: Baar.

Cunningham, H., V. Tablan, A. Roberts, and K. Bontcheva (2013, 02). Getting more out of biomedical documents with gate's full lifecycle open source text analytics. *PLOS Computational Biology 9*(2), 1–16.

de Castilho, R. E., C. Biemann, I. Gurevych, and S. M. Yimam (2014). WebAnno: a flexible, web-based annotation tool for CLARIN. In *Proc. of CAC'14*, Utrecht, Netherlands.

Debus, F. (2012). *Namenkunde und Namengeschichte. Eine Einführung*. Grundlagen der Germanistik. Berlin: Erich Schmidt Verlag.

Dipper, S., M. Götze, and M. Stede (2004). Simple annotation tools for complex annotation tasks: an evaluation. In *Proceedings of the LREC Workshop on XML-based Richly Annotated Corpora*, LREC 2004, pp. 54–62.

Fang, M., Y. Li, and T. Cohn (2017). Learning how to active learn: A deep reinforcement learning approach. In *Proc. of the EMNLP*, pp. 595–605. Association for Computational Linguistics.

Fellbaum, C. (Ed.) (1998). *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press.

Freeman, J. B. (1991). *Dialectics and the Macrostructure of Arguments, A Theory of Argument Structure*. Number 10 in Studies of Argumentation in Pragmatics and Discourse Analysis. Berlin: de Gruyter Mouton.

Gleim, R., S. Eger, A. Mehler, T. Uslu, W. Hemati, A. Lücking, A. Henlein, S. Kahlsdorf, and A. Hoenen (2019). Practitioner's view: A comparison and a survey of lemmatization and morphological tagging in German and Latin. *Journal of Language Modeling*. Forthcoming.

Gleim, R., A. Mehler, and A. Ernst (2012). SOA implementation of the eHumanities Desktop. In *Service-oriented Architectures (SOAs) for the Humanities: Solutions and Impacts. Proc. of the Joint CLARIN-D/DARIAH Workshop at Digital Humanities Conference 2012*, pp. 24–29.

Götz, T. and O. Suhre (2004). Design and implementation of the UIMA Common Analysis System. *IBM Systems Journal 43*(3), 476–489.

Gries, S. T. and A. L. Berez (2017). Linguistic annotation in/for corpus linguistics. In N. Ide and J. Pustejovsky (Eds.), *Handbook of Linguistic Annotation*, pp. 379–409. Dordrecht: Springer Netherlands.

Helfrich, P., E. Rieb, G. Abrami, A. Lücking, and A. Mehler (2018). TreeAnnotator: Versatile Visual Annotation of Hierarchical Text Relations. In *Proc. of LREC 2018*, Miyazaki, Japan.

Hemati, W., T. Uslu, and A. Mehler (2016). TextImager: a Distributed UIMA-based System for NLP. In *Proc. of COLING 2016 System Demonstrations*. Federated Conference on Computer Science and Information Systems.

Kamianets, W. (2000). Zur Einteilung der deutschen Eigennamen. *Grazer Linguistische Studien 54*, 41–58.

Kett, A., G. Abrami, A. Mehler, and C. Spiekermann (2018). Resources2City Explorer: A system for generating interactive walkable virtual cities out of file systems. In *The 31st Annual ACM Symposium on User Interface Software and Technology Adjunct Proceedings*, UIST '18 Adjunct, pp. 123–125.

Klie, J.-C., M. Bugert, B. Boullosa, R. E. de Castilho, and I. Gurevych (2018, Juni). The inception platform: Machine-assisted and knowledge-oriented interactive annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pp. 5–9. Association for Computational Linguistics.

Mani, I. and J. Pustejovsky (2004). Temporal discourse models for narrative structure. In *Proceedings of the 2004 ACL Workshop on Discourse Annotation*, DiscAnnotation '04, Stroudsburg, PA, USA, pp. 57–64. Association for Computational Linguistics.

Mann, W. and S. Thompson (1988). Rhethorical structure theory: Towards a functional theory of text organization. *Text 8*(3), 243–281.

Moro, A., A. Raganato, and R. Navigli (2014). Entity Linking meets Word Sense Disambiguation: a Unified Approach. *Transactions of the Association for Computational Linguistics (TACL) 2*, 231–244.

Nagel, S. (2008). *Lokale Grammatiken zur Beschreibung von lokativen Sätzen und ihre Anwendung im Information Retrieval*. Ph. D. thesis, Ludwig-Maximilians-Universität München.

O'Donnell, M. (1997). RST-Tool: An RST analysis tool. In *Proc. of ENLG 1997*.

O'Donnell, M. (2000). RSTTool 2.4 – A markup tool for rhetorical structure theory. In *INLG'2000 Proceedings of the First International Conference on Natural Language Generation*, pp. 253–256.

Pinto, A., H. G. Oliveira, and A. O. Alves (2016). Comparing the Performance of Different NLP Toolkits in Formal and Social Media Text. In M. Mernik, J. P. Leal, and H. G. Oliveira (Eds.), *5th Symposium on Languages, Applications and Technologies (SLATE'16)*, Volume 51 of *OpenAccess Series in Informatics (OASIcs)*, Dagstuhl, Germany, pp. 3:1–3:16. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

Spiekermann, C., G. Abrami, and A. Mehler (2018, may). VAnnotatoR: a Gesture-driven Annotation Framework for Linguistic and Multimodal Annotation. In J. Pustejovsky and I. van der Sluis (Eds.), *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France. European Language Resources Association (ELRA).

Stede, M. (2007, 01). *Korpusgestützte Textanalyse : Grundzüge der Ebenen-orientierten Textlinguistik*. Narr Studienbücher.

Stenetorp, P., S. Pyysalo, G. Topić, T. Ohta, S. Ananiadou, and J. Tsujii (2012). BRAT: a web-based tool for NLP-assisted text annotation. In *Proc. ofEACL 2012*, Avignon, France, pp. 102–107.

Uslu, T., A. Mehler, and D. Baumartz (2019). Computing Classifier-based Embeddings with the Help of text2ddc. In *Proceedings of the 20th International Conference on Computational Linguistics and Intelligent Text Processing, (CICLing 2019)*, CICLing 2019.

Vasil'eva, N. (2011). Die Terminologie der Onomastik, ihre Koordinierung und lexikographische Darstellung. In K. Hengst and D. Kremer (Eds.), *Namenkundliche Informationen*, Volume 99/100, pp. 31–45. Leipzig: Leipziger Universitätsverlag.

Wilcock, G. (2017). The evolution of text annotation frameworks. In N. Ide and J. Pustejovsky (Eds.), *Handbook of Linguistic Annotation*, pp. 193–207. Dordrecht: Springer Netherlands.

Wither, J., S. DiVerdi, and T. Höllerer (2009). Annotation in outdoor augmented reality. *Computers & Graphics 33*(6), 679–689.

SceneML: A Proposal for Annotating Scenes in Narrative Text

Robert Gaizauskas
University of Sheffield
r.gaizauskas@sheffield.ac.uk

Tarfah Alrashid
University of Sheffield
ttalrashid1@sheffield.ac.uk

Abstract

We propose a scheme to annotate *scenes* in narrative texts. Adopting the widely held definition that a scene is a structural unit in narrative discourse where time, location and characters do not change, we propose: (1) a conceptual model for semantic annotation of scenes and relevant related semantic and textual elements and relations in narratives and (2) a concrete XML-based syntax for scene annotation. We illustrate our ideas via the annotation of an extended example and discuss various issues that have been and remain to be addressed.

1 Introduction

In this paper we put forward an initial proposal for a scheme to annotate *scenes* in narrative texts. Once the scheme is stable we aim to annotate sufficient text to enable a high accuracy classification model to be trained to carry out scene segmentation automatically.

There are several good reasons for doing this. First, there is considerable interest within the field of narratology in studying the structure of stories. A program that could automatically identify scene boundaries in stories would be an invaluable tool, allowing large-scale analysis and quantification of the number, type and length of scenes in different authors' writing.

Secondly, there has been a considerable amount of research integrating vision and language for different tasks, including:

- automatic text illustration or story-picturing;
- aligning books with movies; and,
- automatic generation of image descriptions.

Text illustration, or *story-picturing*, is the task of associating a relevant image with a given piece of text. This problem has been addressed by a number of researchers (Joshi et al., 2004; Feng and Lapata, 2010; Agrawal et al., 2011), but their focus has been on finding suitable images for a given piece of text. However, in illustrating a whole story, the story first needs to be segmented into scenes, as these are the appropriate units to have an associated illustration. Thus a system that could automatically segment a story into scenes, rather then relying on manual segmentation or purely structural information (paragraph or chapter boundaries), would be highly beneficial in such an application setting.

Aligning books with movies is the task of matching segments of a book with shots or sequences of shots in a film version of the book. Zhu et al. (2015) describe a system for tackling this task, which they conceive as a sentence to shot or or sentence sequence to shot sequence task. While this is one way to tackle the broader problem, there may also be merit in aligning scenes in books with scenes in movies, rather than starting at the lower level of sentence to shot alignment. Doing this however, again presupposes a notion of scene and the ability to automatically segment both books and movies into scenes.

Image description is the task generating appropriate linguistic descriptions of image content. Considerable work has been done in this area (Kulkarni et al., 2011; Yang et al., 2011; Dai et al., 2017). While object detection capabilities have improved immensely in the past 5 years, image description involves more than simply listing object types in the image (Wang and Gaizauskas, 2016): choosing which objects to mention is important and this is at least in part dependent on the scene type. Models of scene types and of what object types are frequently mentioned in which scene type descriptions are likely to contribute significantly to improving image description. A likely source of such models is narrative accounts which contain descriptions of scenes, including the objects and actions found within them. However, mining them requires the capability to automatically segment narratives into scenes.

Finally, automatic narrative generation systems, e.g. Callaway and Lester (2002), could benefit from a large corpus of scene-segmented narratives to train aspects of their models.

This paper is structured as follows. Section 2 gives a brief overview of some definitions of *scene* from different areas of research and discusses work related to scene segmentation. Section 3 presents our framework for the annotation of scenes in narrative text. Section 4 provides some examples of our annotation scheme in practice and Section 5 discusses some of the choices we made in specifying the annotation scheme as well as some outstanding issues and challenges. Finally, section 6 briefly summarises our contribution and provides suggestions for future work.

2 Related Work

2.1 Scenes

The notion of scene varies, depending on the context and the area of research in which it is being used. Here, we present an overview of some of the definitions of scene that have been proposed. The one we adopt in this paper will be discussed in the next section.

The first set of authors are interested in the idea of scene as used in the narrative or dramatic art, whether from a computational or purely literary perspective. Callaway and Lester (2002) state that a scene is “a segment of the narrative that is contiguous in time, location and characters”. Changes in any of those three elements (i.e. when characters change their location; when the narrative switches to other characters) signal a scene change. Kozima and Furugori (1994), in their paper “Segmenting Narrative Text into Coherent Scenes”, define a scene as a piece of text that has common characteristics with movie scenes in that they both have characters and objects in a certain situation that includes a specific time, place and background. From the perspective of drama, Polking (1990) defines a scene as “a division within an act of a play, indicated by a change of locale, an abrupt shift in time or the entrance or exit of a major character”.

Cutting (2014), who studied event/scene segmentation in movies, states that a scene is “a medium-size chunk found in all the narrative arts and often synonymous with the concept of an event”. Dunne (2017) claims that the physical life of a setting presents the truth about the characters in the story or in the scene. The setting is defined by physical life, time and physical objects; it is about knowing where and when the scene is taking place. For example, actions that occur at night differ from those that occur during the day, and outdoor settings are different from indoor settings. Physical life helps the reader imagine the setting; therefore, it brings visual power to the scene.

By contrast, a second group of researchers is interested in what in vision research is referred to as “scene understanding”. Xiao et al. (2010), for example, define a scene as any place or location in which a person can take action or navigate. Scenes are often either related to or associated with specific actions (e.g. sleeping in a bedroom or reading in a library), and they are related to the space’s visual features. The environment is defined by its size and shape (e.g. a narrow corridor is for walking), by its material (e.g. snow, grass, wood) and by its objects (e.g. table and chairs). They build a database of 908 scene categories with associated images by manually selecting, and then merging where appropriate, common nouns from Wordnet that could reasonably be used to complete sentences like “I am in a *place*”, or “Lets go to the *place*”. Note this use of “scene” is similar to the use of term “setting” by authors interested in narrative, i.e. focussed on location and time, but not on characters.

2.2 Scene Segmentation

The task of segmenting narratives into scenes has not been widely investigated (as opposed to the more general task of segmenting text of any genre into topically coherent subsections, which has been much more widely addressed – see e.g. Koshorek et al. (2018) for some recent work in this area). Kozima and Furugori (1994) introduce a method of segmenting narrative text into coherent scenes, suggesting that it could be used to help resolve anaphora and ellipsis inside a scene. They define a scene as a set of continuous sentences that are coherent. A scene in a text is like a movie scene in that it describes objects, such as characters or properties, in a certain time, place and background. Scene segmentation is done using a Lexical Cohesion Profile (LCP), which was first proposed by Kozima (1993), to discover boundaries of scenes in narrative text. It relies on the idea that coherent text is lexically cohesive (Morris and Hirst, 1991; Halliday and Hasan, 1976), with local cohesiveness leading to local coherence. An LCP keeps a record of lexical cohesion between words inside a window, by moving a window of a certain size (i.e. 50 words) word by word and measuring the lexical cohesion between the words each time the window is moved. There is a relationship between LCPs and change of scene; when a window is inside a scene, the LCP value is typically high and words tend to be lexically cohesive; however, when a window crosses a scene boundary, the LCP value decreases and the words vary lexically. Thus, LCPs can help identify scene changes by detecting the valleys that are the minimum points. The authors claim that LCPs can be an indicator of scene changes, as validated by comparison to human judgements. However, lexical cohesiveness and text coherence do not always correlate. Sometimes, lexical cohesiveness is high on incoherent text. In addition, in some cases some texts are coherent but are not lexically cohesive.

Kauchak and Chen (2005) also investigate the segmentation of narrative documents. However, they do not tackle the problem of scene segmentation. Their study used two narrative books: “Biohazard” by Ken Alibek and ”The Demon in the Freezer” by Richard Preston. The authors of these books themselves segmented the books into sections, and these sections were then used as true segment boundary locations by Kauchak and Chen. However, there is no discussion of what the authors’ basis for segmentation into sections was. Segmentation was then treated as a binary classification problem where each sentence boundary is a potential segmentation point. Various supervised learning algorithms and features were explored to automate the task.

Cutting (2014) studies the correlation between viewer segmentation of movies into events and narrative shifts in time, location and characters in these movies. He first had viewers manually segment films into events based on their intuitive notions of event. Then he manually analysed 31,000 shot boundaries in the same films and categorised them into 8 categories depending on which combination of time, location or character shifted at the shot boundary. He observed that the more narrative parameters (time, location, character) shifted at a shot boundary, the more likely a shot boundary was to correspond to an event boundary.

In sum, few authors have addressed the problem of automatically segmenting narrative texts into scenes. Authors address segmentation of text but not narrative text; or they address segmentation of narrative text but not into scenes; or they investigate segmentation of narrative into scenes, but in movies and not in text and do it manually. Therefore there is opportunity for novel work to address automatic segmentation of narrative text into scenes. And, in order to support that effort, work is needed to clearly define what scenes are in narrative text and to produce annotated resources based on this definition for training and testing of automatic segmentation systems. The following sections address this requirement.

3 The Annotation Framework

We distinguish the conceptual model that underlies our approach to annotation from the concrete syntax we employ in the annotation scheme (cf., e.g, Pustejovsky et al. (2011)).

3.1 The Conceptual Model

Following the general tendency we observe in the literature (Section 2.1), we shall treat a *scene* as a unit of a story in which the time, location and major characters are coherent, i.e. stay essentially the same. A change in any one these constitutes a change of scene.

Note that a scene is an abstract discourse element and does not exist apart from the narrative of which it forms a part. It comprises a location or setting, a time and one or more characters who participate in actions or events that unfold in the scene. By contrast, each of these things (location, time, character and events) *do* indeed exist in the real or fictive world (or *storyworld* as per narrative theory) in which the narrative is set. But the scene itself is an abstraction away from the potentially infinitely complex detail of that real or fictive world, a cognitive construct that makes finite, focussed narrative possible.

A scene is realised in the textual narrative as one or more *scene description segments (SDS)*. An SDS is a contiguous span of text that, possibly together with other SDSs, expresses a scene. Generally, a scene will consist of a single SDS unless that SDS contains embedded SDSs for other scenes, typically for temporally discontinous scenes (e.g. memories of past scenes or imagined future scenes) or spatially distinct locations that are topologically contained within or connected to the embedding SDS, or if the author is employing the narrative device of rotating between multiple concurrent scenes each of which is advancing a distinct story line (a technique very commonly used in action movies).

Since scenes change when *characters*, *time* or *location* change, it seems a good discipline in annotation to identify what these are for each scene. Furthermore, as we have seen in Section 2.2, others such as Cutting (2014) have found it useful to study types of scene changes. Identifying which of character, time and location changes between scenes will contribute to such studies. We see no need to re-invent the wheel and so are hopeful we can simply adopt the definitions, and annotation standards, for times, locations and spatial entities from Iso-TimeML and Iso-Space[1]. For characters, we propose to adopt the definition and annotation standards for named entities of type person, as developed for the ACE program and recently used in, e.g. the TAC 2018 entity discovery and linking task[2].

These standards are appropriate for annotating *all* mentions of times, locations/spatial entities and persons, in texts. However, we are interested in specific times, locations and persons, namely those which represent the time of, location of and characters of the scene in which they occur. Thus, conceptually, we are interested not just in these entities, but in *relations* between these entities and the narrative construct which is the scene.

3.2 SceneML Elements

SceneML comprises two main element types:

1. *Entities*: scenes, scene description segments (SDSs), locations, times, characters

2. *Relations*: scene-scene narrative progression links; there are other relations – character-scene, location-scene and time-scene relations – but for now we represent these via attributes in entities (see below in Section 5 for a discussion of alternatives and whether this a good way to go).

We give a brief description of each of these here. An extended example that illustrates the use of each is provided in the next section.

Scenes The primary element in SceneML. As indicated above, scenes are a core basic unit in any form of storytelling and involve a coherent setting (location and time) and set of characters, who participate in some form of action and/or dialogue that advances the story. They have as attributes a unique `id`, `time` and `location`. They include a set of `character` sub-elements, since there may be arbitrarily many characters per scene,

[1] `https://www.iso.org/standard/37331.html` and `https://www.iso.org/obp/ui/#iso:std:60779:en`.

[2] See `http://nlp.cs.rpi.edu/kbp/2018/` and `https://www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/english-entities-guidelines-v5.6.6.pdf`.

Scene description segments (SDSs) SDSs are text spans that are the textual components of scenes. Each SDS is a component of exactly one scene, though a scene may comprise many SDS's. An SDS has the following attributes: a unique `id` and a unique `scene_id`, which is the id of the scene which includes the SDS as a component.

Time Times are essentially timex3 elements as proposed in ISO-TimeML. They have an `id` attribute and a text span. We also allow a special time value of `base`, which is the time in the storyworld of the narrated events.

Location Locations are either location or spatial entity elements from ISO-Space. Like times they have an `id` attribute and a text span. Other attributes from the ISO-Space elements may be adopted as we conduct further corpus annotation.

Character Characters are entities of type `person (PER)` as specified in the ACE English Annotation Guidelines for Entities (see note 2 above) with the exception that we *do* allow characters to be animals or non-human, provided they play the role of a character in the narrative. They have an `id` attribute, a `type` attribute and a text span. The `type` attribute used in ACE allows person entity mentions to be sub-typed along various dimensions: whether the mention is generic or specific (`GEN` vs `SPC`), of an individual or a group (`Ind` vs `Group`), and in pronominal, nominal or proper nominal form (`PRO`, `NOM`, `NAM`). References to characters in narrative are specific, generally individual and may take any form. Further consideration is needed as to whether just to include individual person entities in our model, in which case the only sub-typing of interest will relate to the form of the mention, or whether to include a broader set of person entities, e.g. groups of persons, such as teams or sides in a narrated conflict, that may include individual characters. We are inclined to allow groups to be treated as characters, since they can function as such in narratives.

Narrative Progression Links Narrative progression links (`nplinks`) express the type of narrative progression between textually adjacent scenes. For now we identify four types of progression: `sequence`, when one scene follows on from another, e.g. when characters move from one location to another; `analepsis` (or flashback), when we are taken to another, earlier time and possibly other details such as location and characters change as well; `prolepsis` (or flashforward), when we are taken forward in time; `concurrent` when we are taken to another location with different characters, where another thread of the story is developing at the same time as the textually preceding scene. Since scenes may be expressed across several non-contiguous text segments, it is possible that one scene may have an `nplink` with more than one other textually adjacent scene (see scenes 3-5 in Figure 1 below). This could be taken to suggest that `nplinks` should hold between SDSs rather than between scenes. However, we view narrative progression as properly being a relation between units of narrative, i.e. scenes. Furthermore, in the vast majority of cases there is just one SDS per scene. Therefore, we propose that `nplinks` be scene-scene relations. In any case, for multi-segment scenes the information about where in the scene a shift is made to an another scene, and to which other scene, is recoverable from the annotation, should it be required.

4 Example SceneML Annotations

Figure 1 shows the annotation of several pages of a children's story called *Bunnies from the Future* by Joe Corcoran (Corcoran, 2016). Much of the text has been elided and some of the annotations have been simplified to improve readability. Specifically, rather than explicitly annotating times, locations and persons in the text and then using their `id` attribute value in the `scene time` or `location` attribute or in the `character` element, we use the text string itself as the value of the attribute or data element.

Since the text is so elided for space reasons, we provide a short summary of the narrative segment here to aid comprehension. The protagonist (*I*), who is a human child, has been transported to the future

```
<scene id="s1" loc="pod" time="base">
    <character>I</character>
    <character>Skip</character>
    ...
</scene>
<sds id = "sds1" scene = "s1">
As we approached our destination, Skip started to issue instructions to the pod about approach vectors ... I
was about to say something ... when Skip opened the door and
</sds>
<scene id="s2" loc="bubble shaped large room" time="base">
    <character>I</character>
    <character>Skip</character>
    ...
</scene>
<sds id = "sds2" scene_id="s2">
I stumbled out into a blaze of light and noise ... I was in a large room ... It was bubble shaped ... All around
me in the bubble were bunnies ... " ... Right now, we have to take him to Methuselah. ... I was just about to
ask if Skip was coming in with me, but the door had already opened and they were manoeuvring me through.
</sds>
<scene id="s3"  loc="cylindrical room"  time="base">
    <character>I</character>
    <character>Methusaleh</character>
</scene>
<sds  id = "sds3" scene_id="s3">
I found myself in a room that was cylindrical, like the pod only bigger ... There, in front of, or above, me
(zero gravity is so confusing) was the oldest rabbit I had ever seen ... Methuselah then told me about how the
Bunnies from the Future first came into being.
</sds>
<scene id="s4" loc="planet earth"  time="a long time ago">
    <character>humans</character>
    <character>bunnies</character>
</scene>
<sds  id = "sds4" scene_id="s4">
It was after the plants had turned nasty  a long time ago, even for this old bunny  and the story sounded more
like a legend than real history ... The war was lost, but the bunnies never stopped searching for a way to
achieve victory and to reclaim the planet.
</sds>
<sds id = "sds5" scene_id="s3">
What happened to people? Where are they now? I interrupted, rather rudely ... The door behind me swished
open and it was clear that my audience with Methuselah was over. I gave one last look at the ancient bunny,
before shaking my head and making my way, carefully, to the exit ...
</sds>
<scene id="s5" loc="corridor"  time="base">
    <character>I</character>
    <character>Skip</character>
</scene>
<sds  id = "sds6" scene_id="s5">
Skip and the other bunnies were still waiting when I squeezed my way out through the door ...
</sds>
<nplink id = "npl1" type = "sequence" scene1 = "s1" scene2 = "s2"/>
<nplink id = "npl2" type = "sequence" scene1 = "s2" scene2 = "s3"/>
<nplink id = "npl3" type = "analepsis" scene1 = "s3" scene2 = "s4"/>
<nplink id = "npl4" type = "sequence" scene1 = "s3" scene2 = "s5"/>
```

Figure 1: Example SceneML Annotations for Chapter 2 of Corcoran (2016)

by super-smart bunnies, who have been forced to live in space by evil plants who have taken over Earth. In the annotated episode, the protagonist is arriving at the bunnies' "space station" having been flown there in a pod, commanded by *Skip*. He is an object of curiosity to the many bunnies in the multi-roomed space station at which he arrives, but is shortly taken to see *Methusaleh*, an aged rabbit who appears to be the rabbits' leader. Methusaleh recounts some history to the protagonist, explaining how the plants took over the Earth many years ago now and how the bunnies have ever since sought to reclaim it.

5 Discussion

Representing Relations in XML One issue that came up repeatedly in developing the annotation scheme was whether to represent relational information between two entities via an explicit relation element or via an attribute or element associated with one of the entities. So, for example, every `sds` is associated with exactly one `scene`. We have chosen to record this via the `scene_id` attribute of the `sds` element. But it could have been represented in a separate `sds-scene` link entity with two attributes, one for `sds_id` and one for `scene_id`. These representations are informationally equivalent. There may turn out to be processing considerations that favour one over the other; but then the less efficient representation could always be converted to the more efficient by an automated process. So probably visual economy and ease of annotation are the primary considerations. Other places where this problem arises are in character-scene relations. We have chosen to handle this by associated multiple character sub-elements with scenes. But these too could be represented as link relations between character and scene ids.

Scene transition signals Some sentences serve to signal a scene transition. Consider this slight variant of one of our example sentences above: *Skip opened the door and I stumbled from the pod out into a blaze of light and noise*. Two questions arise here: (1) should such sentences be included in the first scene, the second scene, in both scenes or in neither? or be split somehow in the middle? (2) should we annotate them, as e.g., `scene_transition_signals`, much the way that temporal and spatial signals (e.g. *before* or *in front of*) are annotated in ISO-TimeML and ISO-Space? We are leaning towards annotating them as signals separate from each scene, which will have the advantage of assisting supervised learning algorithms to identify scene transition markers; but we have not reviewed sufficient data yet to make a robustly evidenced recommendation.

6 Conclusion

In this paper we have proposed an annotation framework – a conceptual model and XML syntax – for annotating scenes in narrative texts. The definition of scene is based on the relatively widely shared view in narrative studies that scenes change whenever time, location or principal characters shift. Following this view, we propose an abstract scene entity, which is realised in text via one or more scene description segments, contiguous sequences of sentences describing the action and dialogue in a scene. Scenes have associated time, location and character information and we propose to adopt previously developed annotation standards for these things. We illustrated our proposal via an extended example and discussed various issues relating to it.

Future work will take the form of an iterative cycle of annotating texts (expanding the type of texts covered not just the quantity) and refining the annotation specification and guidelines. Of course some text will need double annotation and inter-annotator agreement will be assessed; we will also assess the feasibility of crowd sourcing annotation and of using texts with pre-existing scene annotations (e.g. plays, screen plays, etc.). As noted in the introduction, our aim is to annotate a large enough corpus to be able to train and evaluate an automatic scene segmenter. This will then help enable a range of applications, including narrative analysis tools, book to movie alignment, image description and narrative generation.

Acknowledgements The authors would like thank the ISA-15 reviewers whose comments have helped improve the paper. The second author would like to acknowledge support from the University of Jeddah in the form of a PhD studentship.

References

Agrawal, R., S. Gollapudi, A. Kannan, and K. Kenthapadi (2011). Enriching textbooks with images. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, New York, NY, USA, pp. 1847–1856. ACM.

Callaway, C. B. and J. C. Lester (2002, August). Narrative prose generation. *Artif. Intell. 139*(2), 213–252.

Corcoran, J. (2016). *Bunnies from the Future*. www.freekidsbooks.org.

Cutting, J. E. (2014, jun). Event segmentation and seven types of narrative discontinuity in popular movies. *Acta Psychologica 149*, 69–77.

Dai, B., S. Fidler, R. Urtasun, and D. Lin (2017). Towards diverse and natural image descriptions via a conditional gan. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2970–2979.

Dunne, W. (2017). *The dramatic writer's companion: tools to develop characters, cause scenes, and build stories*. University of Chicago Press.

Feng, Y. and M. Lapata (2010). Topic models for image annotation and text illustration. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 831–839. Association for Computational Linguistics.

Halliday, M. A. K. and R. Hasan (1976). *Cohesion in English*. Longman.

Joshi, D., J. Z. Wang, and J. Li (2004). The story picturing engine: finding elite images to illustrate a story using mutual reinforcement. In *Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval*, pp. 119–126. ACM.

Kauchak, D. and F. Chen (2005). Feature-based segmentation of narrative documents. In *Proceedings of the ACL Workshop on Feature Engineering for Machine Learning in NLP*, pp. pages 3239.

Koshorek, O., A. Cohen, N. Mor, M. Rotman, and J. Berant (2018, June). Text segmentation as a supervised learning task. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, New Orleans, Louisiana, pp. 469–473. Association for Computational Linguistics.

Kozima, H. (1993). Text segmentation based on similarity between words. In *Proceedings of the 31st annual meeting on Association for Computational Linguistics*, pp. 286–288. Association for Computational Linguistics.

Kozima, H. and T. Furugori (1994). Segmenting narrative text into coherent scenes. *Literary and Linguistic Computing 9*(1), 13–19.

Kulkarni, G., V. Premraj, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg (2011). Baby talk: Understanding and generating image descriptions. In *Proceedings of the 24th CVPR*. Citeseer.

Morris, J. and G. Hirst (1991). Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational linguistics 17*(1), 21–48.

Polking, K. (1990). . *Writing A to Z: The terms, procedures, and facts of the writing business defined, explained, and put within reach. Cincinnati, OH*. Writer's Digest Books. ISBN 0-89879-435-8.

Pustejovsky, J., J. Moszkowicz, and M. Verhagen (2011). ISO-Space: The annotation of spatial information in language. In *Proceedings of the Sixth Joint ACL-ISO Workshop on Interoperable Semantic Annotation*, pp. 1–9.

Wang, J. and R. Gaizauskas (2016). Don't mention the shoe! a learning to rank approach to content selection for image description generation. In *Proceedings of the 9th International Natural Language Generation Conference (INLG16)*.

Xiao, J., J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba (2010, jun). SUN database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3485–3492. IEEE.

Yang, Y., C. L. Teo, H. Daumé III, and Y. Aloimonos (2011). Corpus-guided sentence generation of natural images. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 444–454. Association for Computational Linguistics.

Zhu, Y., R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, Washington, DC, USA, pp. 19–27. IEEE Computer Society.

Annotating and Recognising Visually Descriptive Language

Tarfah Alrashid
Department of Computer Science,
University of Sheffield, UK
ttalrashid1@sheffield.ac.uk

Josiah Wang
Department of Computing,
Imperial College London, UK
jw@josiahwang.com

Robert Gaizauskas
Department of Computer Science,
University of Sheffield, UK
r.gaizauskas@sheffield.ac.uk

Abstract

We propose recognising "visually descriptive language" (VDL) as an interesting and potentially useful task for researchers working in the field of Vision & Language integration. Adopting the definition of VDL proposed by Gaizauskas et al. (2015), that VDL is language that asserts propositions whose truth can be confirmed through visual sense alone, we investigate the specific task of classifying sentences as wholly, partially or not at all visually descriptive. We discuss the annotation of VDL on several texts, and report results on automatic classifiers trained on the annotation, showing that the task can be performed at around 79% accuracy, suggesting that this is a potentially fruitful avenue for further research.

1 Introduction

Recent years have seen the introduction of various joint Vision and Language (V&L) tasks such as image and video description generation or captioning (Bernardi et al., 2016; Kuznetsova et al., 2012; Kulkarni et al., 2011; Fang et al., 2015; Karpathy and Fei-Fei, 2015; Vinyals et al., 2015), visual question answering (Antol et al., 2015; Yu et al., 2015; Zhu et al., 2016), visual dialog (Das et al., 2017; Chattopadhyay et al., 2017), among others. The approaches taken to many of these tasks rely on artificially constructed datasets, where, e.g., image descriptions of a prescribed type are collected from human annotators, frequently via crowd-sourcing, for a selected group of images.

There are several limitations to such approaches. First, it is unclear whether they will ever generalise sufficiently to transfer to real world data substantially different from the training data in these datasets. Secondly, while joint models have indisputable strengths, by training on image-description pairs only they neglect the huge amount of unpaired data that is available in the form of undescribed images and visually descriptive language in the absence of images.

In this paper, we propose the challenging task of recognising *visually descriptive language* regardless of whether it occurs with an associated image or in a text such as a novel, newspaper or travel article, or biography without images. This task is challenging as it requires a deep and nuanced understanding of texts beyond surface-level understanding, and also needs sufficient world knowledge to ground texts to the visual world. The task is an interesting challenge in and of itself as it can potentially give rise to deeper insights into the concept of visualness and how it corresponds to its usage in language. Furthermore, being able to recognise visually descriptive language may prove useful for different applications, for example for developing (i) models that can exploit knowledge about co-presence or dependence of object types in particular settings to assist in object recognition; (ii) models for content selection when deciding what to mention in images or in particular settings.

We first briefly review two definitions of "visually descriptive language" that have appeared in the literature (§2). Choosing one of them, we go on to describe the annotation of visually descriptive language on several text instances based on the definition as a guide (§3). We also present preliminary work on automatically recognising visually descriptive information at the sentence level using supervised learning models (§4 and §5) – the first work to do so. Results are sufficiently encouraging to suggest that automatically recognising visually descriptive language is a feasible task.

2 Defining Visually Descriptive Language

We are aware of only two attempts to define visually descriptive language. Dodge et al. (2012) begin by assuming an image and an associated natural language description and define visual language in the description as "A piece of text is visual (with respect to a corresponding image) if you can cut out a part of that image, paste it into any other image, and a third party could describe that cut-out part in the same way" (Dodge et al., 2012, pp. 763).

Another definition of visually descriptive language (VDL) was proposed by (Gaizauskas et al., 2015, pp. 11–12):

> A text segment is *visually descriptive* iff it asserts one or more propositions about either (a) a specific scene or entity whose truth can be confirmed or disconfirmed through direct visual perception (e.g. (1)), or (b) a class of scenes or entities whose truth with respect to any instance of the class of scenes or entities can be confirmed or disconfirmed through direct visual perception (e.g. (2)).
>
> 1. *John carried a bowl of pasta across the kitchen and placed it on the counter.*
> 2. *Tigers have a pattern of dark vertical stripes on reddish-orange fur with a lighter underside.*
> 3. *Maria is thinking about what the future holds for her.* (Not VDL)

This definition is further elaborated in Gaizauskas et al. (2015) where what *assert one or more propositions* and *direct visual perception* mean are explored in more detail [1].

Unlike Dodge et al. (2012) who define visual text at sentence level, Gaizauskas et al. (2015) define visually descriptive language at the level of *text segments* which could be a phrase, a sentence or a sequence of sentences. Some text segments (e.g. *the tall, well-educated man*), however, may comprise subsegments that express visually confirmable (*tall*) and not visually confirmable (*well-educated*) properties. They term such cases, where the visually descriptive elements are non-contiguous, as *impure* VDL (IVDL). More examples of such cases are shown in Figures 1 and 2 as dotted lines connecting IVDL subsegments.

Gaizauskas et al. (2015) also provide guidelines for annotating many difficult cases including: metaphors; words with mixed visual and aural meanings (*shout*); temporal adverbials (*always*); intentional contexts; hypotheticals, modals, counterfactuals and subjunctives; statements of purpose; imperative and interrogative sentences; participial phrases; indirect speech. We refer interested readers to Gaizauskas et al. (2015) for a more elaborate discussion on these difficult cases, as well as some example disagreements.

Dodge et al. (2012) are primarily concerned with filtering non-visually descriptive language from image captions (e.g. language that refers to the photographer). While they suggest their definition could be applied more generally to any text, they do not pursue this at length. Furthermore their focus is on noun phrases that provide object designations rather than on visual language more generally. For these reasons we adopt the definition proposed by Gaizauskas et al. (2015) in this paper.

[1] For instance, a definite NP that predicates a visually confirmable property of an entity type and whose referential success depends upon the truth of an associated presupposition is deemed visually descriptive. E. g. *the green door* successfully refers only if *There exists a door and that door is green*. Hence *the green door* is visually descriptive. By contrast *the door* in *The door belongs to Jim* is not, since there is no visually confirmable property asserted of any entity. This is not to say that *door* is not visual, but that there is nothing visually descriptive asserted here. The decision not to annotate bare nouns reflects the pragmatic consideration that lists of physical, i.e. potentially visual, entity types already exist (e.g. in WordNet) and nothing is to be gained from annotating mention of these in contexts where nothing visual is asserted of them.

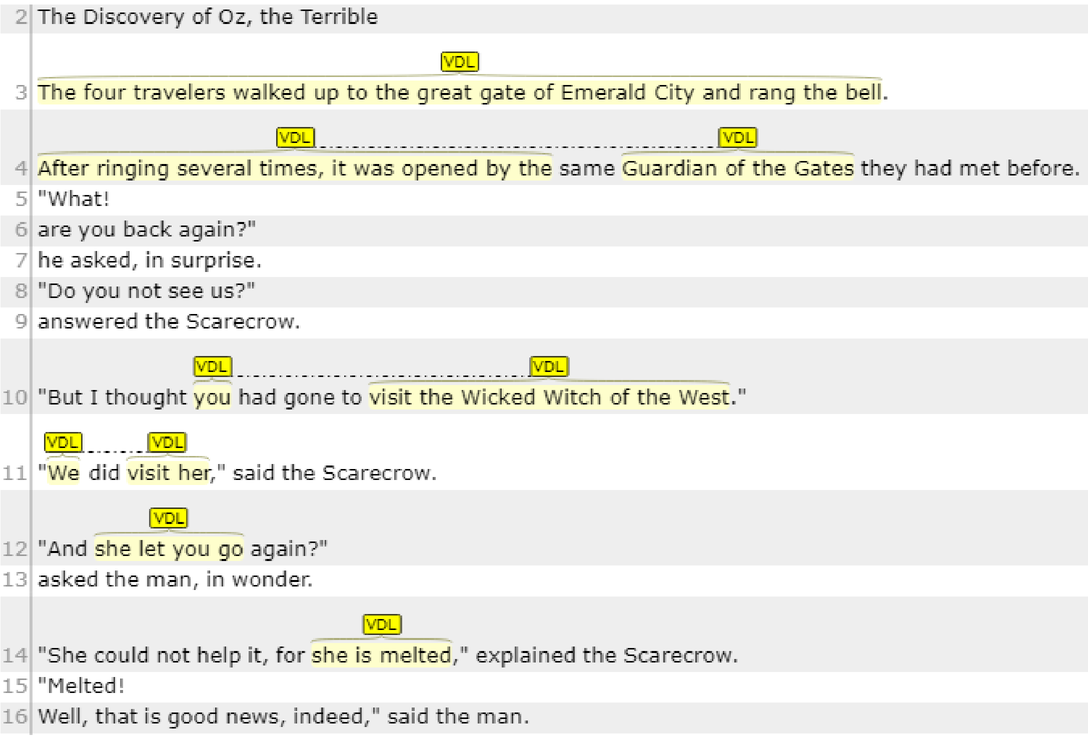

Figure 1: An example segment-level annotation of VDL from one annotator for The Wonderful Wizard of Oz, annotated using the brat rapid annotation tool.

3 Data: VDL Annotated Corpus

Gaizauskas et al. (2015) pilot their annotation scheme for VDL by having two to three annotators carry out segment-level annotation on two randomly selected chapters from The Wonderful Wizard of Oz (WOZ) and six samples from five categories in the Brown Corpus (two news reports, one biography and three novels), resulting in a total of 173 sentences for WOZ and 779 sentences for the Brown Corpus samples. The authors assumed the texts to be visually descriptive, and found that the assumption holds for stories, but less so for news reports and biographies. Adventure novels are also more visually descriptive than romance, with the latter focussing more on the mental states and processes of characters in the story. The authors also provide inter-annotator agreement statistics for their corpus, reporting a segment-level intersection over union (IoU) score ranging from 0.43 to 0.73, and sentence-level (§3.1) κ scores ranging from 0.70 to 0.87.

In this paper, we extend the dataset of Gaizauskas et al. (2015). More specifically, we augmented the WOZ corpus to cover 8 chapters (odd numbered chapters from 1 to 15), all annotated at segment level by two external annotators who are not directly involved in this research. The annotators performed the annotation using the brat rapid annotation tool[2]. Our extended WOZ corpus contains 916 sentences in total, with each chapter ranging from 52 to 203 sentences. Note that this extended WOZ corpus also includes the same two chapters (chapters 7 and 9) from Gaizauskas et al. (2015), thus there are now five annotators for these two chapters. Combining the extended WOZ and the Brown Corpus samples gives us a total of 1,695 sentences. Figure 1 shows an example segment-level annotation from the WOZ corpus, where VDL segments are highlighted, and where visually descriptive subsegments are connected with a dotted line to form a non-contiguous IVDL segment (see §2).

Table 1 shows the statistics for the extended WOZ corpus across two annotators.Column $|S|$ shows the number of sentences in the chapter. Columns **VDL** and **IVDL** show the average number of segments across two annotators marked as pure and impure VDL respectively. Across five annotators, the average number of pure and impure VDL is 59.2 and 29.0 respectively for Chapter 7, and 41.0 and 15.8 for

[2]`http://brat.nlplab.org`

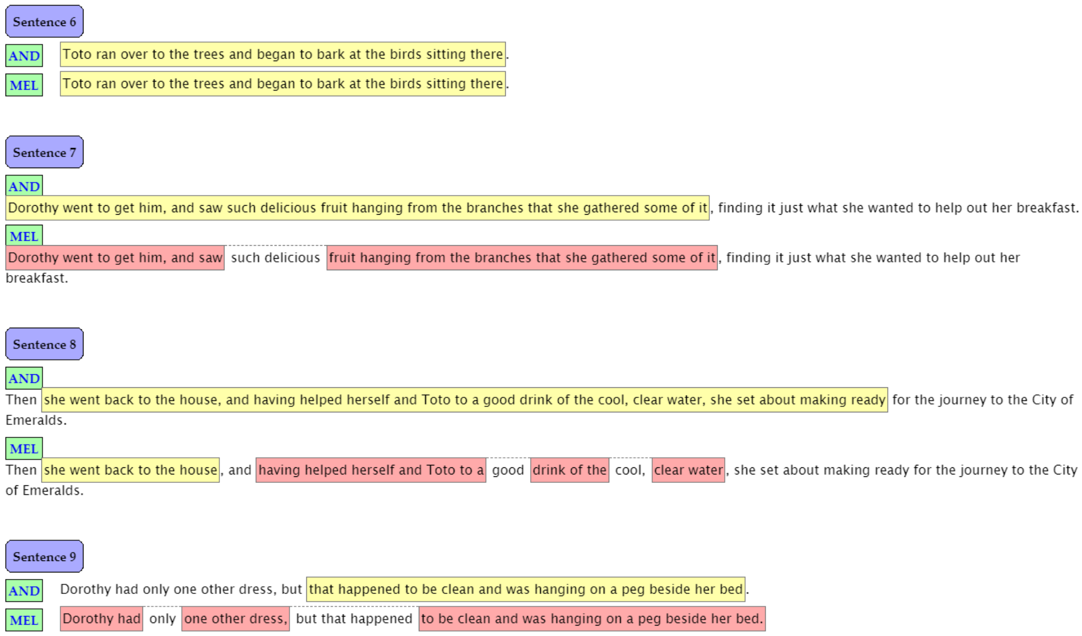

Figure 2: A few example annotations from The Wonderful Wizard of Oz, where we compare annotations from two different annotators (AND and MEL). Yellow segments indicate VDL, while red segments connected by dotted lines indicate impure VDL (IVDL).

Chapter 9. **IoU** is the word-level intersection-over-union score ranging from 0.40 to 0.73, which gives us a rough approximation of the segment-level agreement. As an illustrative example, Figure 2 compares the annotations of two annotators for four sentences from the WOZ Corpus, where we see how the annotators can disagree on some fine-grained elements, e.g. *delicious* in the second sentence.

3.1 Sentence-level Annotation

Gaizauskas et al. (2015) propose two annotation tasks: one at *segment* level and one at *sentence* level. In segment-level annotation, each VDL segment within each sentence S is annotated. In sentence-level annotation, each sentence S is assigned a label: (i) **0** if it does not contain any VDL (*not VDL*); (ii) **1** if it consists entirely of VDL (*fully VDL*); (iii) **2** if it contains one or more proper subsegments that are VDL, but also contains segments that are not (*partially VDL*). In this paper, we concentrate on the latter task, i.e. sentence-level annotation.

Like Gaizauskas et al. (2015), we obtain sentence-level annotations for the classification task (§4) by inferring from the segment-level annotations. If a marked segment spans the whole sentence, then we annotate the sentence as **1** (fully VDL). If a sentence does not contain any marked segments, we annotate the sentence as **0** (not VDL). If a marked segment almost spans the whole sentence, we annotate the sentence as **1** if the unmarked characters at the beginning or end of the sentence consist entirely of punctuation, white space and/or stop words (*so*, *but*, *and*, *the*, *when*, etc.), and as **2** (partially VDL) otherwise. If a sentence consists of multiple segments, we annotate it as **1** if all unmarked segments are made up of punctuation, white space and stop words, and **2** otherwise.

Tables 1 shows the statistics for the inferred sentence-level annotations for the two annotators. Columns **S=1** and **S=2** show the average proportion of sentences labelled as fully VDL (**1**) and partially VDL (**2**) respectively. Columns **%Agree** and κ show the inter-annotator agreement at sentence level, where κ is the Cohen's kappa coefficient. For the two chapters with five annotators, the average agreement is 0.79 and 0.83 for Chapters 7 and 9 respectively, and the average pairwise kappa is 0.65 and 0.71 respectively. The annotations and the detailed annotation guidelines are available online[3].

[3]`http://vdlang.github.io/`

Chapter	$\|S\|$	S=1	S=2	VDL	IVDL	%Agree	κ	IoU
1	59	0.19	0.54	40	14	0.68	0.47	0.72
3	124	0.12	0.40	57	18	0.78	0.64	0.73
5	112	0.07	0.43	48	21	0.72	0.52	0.56
7	95	0.14	0.54	72	18	0.77	0.60	0.72
9	78	0.15	0.43	43	14	0.79	0.67	0.66
11	203	0.15	0.40	99	33	0.73	0.56	0.70
13	52	0.06	0.67	41	15	0.77	0.51	0.73
15	193	0.03	0.30	52	26	0.72	0.41	0.40

Table 1: Statistics of the extended WOZ corpus (averaged across two annotators). $|S|$ is the number of sentences. **S=1** is the average proportion of sentences labelled at sentence level as fully VDL, and **S=2** for partially VDL. **VDL** and **IVDL** are the average number of segments marked as VDL/IVDL. **%Agree** and Cohen's κ show the inter-annotator agreement at sentence level, and **IoU** the agreement at segment level. Please see main text for more details.

4 Sentence-level Classification: Representations

We investigate the task of automatically classifying VDL at sentence level, i.e. annotating a sentence as either **0** (not VDL), **1** (fully VDL) or **2** (partially VDL). We focus on machine learning techniques that learn from our human-annotated data (§3.1) and assume, for now, that sentences are independent of each other, i.e. we treat each sentence as an independent classification task.

In this paper, we focus on exploring different sentence representations supplied to our VDL classifiers. We divide the representations into two groups: (i) representations that explicitly encode 'visualness'; (ii) representations using word tokens in the sentences. We also investigate combining different features from within or across the two groups.

4.1 Explicit visualness representations

Our first group of representations explicitly encodes whether components of a sentence are 'visual'. We investigate using two knowledge bases to infer a sentence's 'visualness': **VerbNet** (Schuler, 2005) and **WordNet** (Fellbaum, 1998). The main idea behind this method is to exploit the knowledge bases for identifying the 'visualness' of elements of the sentence to be classified, e.g. are the main verb, subject and/or object 'visual'?

VerbNet: Our first representation is a 2-dimensional *binary* vector, indicating (i) whether a verb is detected in the sentence; and (ii) whether the verb is 'visual'. We extract the main verb of the sentence by taking the root of its dependency tree generated using the neural network based dependency parser (Chen and Manning, 2014) as implemented in Stanford CoreNLP (Manning et al., 2014). To determine whether the verb (if it exists) is 'visual', we query VerbNet (Schuler, 2005) with the verb to obtain its closest VerbNet class. A verb is considered 'visual' if its VerbNet class can be found in our manually constructed list of 'visual' VerbNet classes.

WordNet: Our second representation is similar to the first, except that we now also consider the subject and object of the sentence in addition to the main verb, and use WordNet to infer the 'visualness' of the main verb, subject and object of the sentence. Like the VerbNet representation, we extract the main verb and the subject and object associated with the main verb from the dependency tree of the sentence. Thus, our WordNet representation is a 4-dimensional vector encoding (i) whether a main verb is detected in the sentence[4]; (ii) whether the subject associated with the main verb, if any, is 'visual'; (iii) whether the

[4]We also tried encoding the presence/absence of the subject and object but found that it made no difference to the final score.

object, if any, is ‘visual’; (iv) the synset label of the root hypernym for the main verb. The first three are binary features, and the final is a categorical string label. At test time, we set the final feature to a ‘0’ string if no verbs are detected or if the root hypernym synset is unseen at training time. For encoding the ‘visualness’ of the subject and object, we query WordNet for the best matching synset, and assume that the subject/object is ‘visual’ if the lemma ‘physical’ occurs in any of its inherited hypernyms.

4.2 Representations using word tokens

In this second group of representations, we do not explicitly infer or encode the ‘visualness’ of words, but instead use the word tokens from the sentence as a features. We explore two approaches: (i) a tf-idf weighted bag-of-words representation; (ii) average word embeddings. The intuition is that the ‘visualness’ of the sentences will be implicitly captured by the sentence classifier.

tf-idf: We experiment with representing a sentence as a bag of words, i.e. each sentence is represented as a vector of term frequencies (*tf*) weighted with the inverse-document frequency (*idf*). *idf* is computed over all the sentences in the dataset.

Word embeddings: We also explore a word embedding-based model that represents words in the sentence in a distributional vector space. In such approaches, words often used in the same context will be close in the semantic space. As word embeddings, we use 300-dimensional GloVe vectors (Pennington et al., 2014) from spaCy[5], which have been trained on a set of web documents from Common Crawl[6]. A sentence-level vector is produced by averaging the word vectors for each word in the sentence that is not a stop word.

4.3 Combining features

We also attempted to concatenate the features from our **WordNet** approach and **tf-idf** (**WordNet+tf-idf**). We also explore concatenating the **word embedding** vector with the **tf-idf** vector (**Embedding+tf-idf**).

5 Experiments and Results

In this section, we report and discuss the results of our experiments on the VDL sentence-level classification task, comparing three supervised classifiers with the different representations proposed in §4. The three classifiers that were explored and tested are: (i) a weighted k-nearest neighbours (kNN) classifier with a Euclidean distance measure (we use $k = 5$ in our experiments); (ii) support vector machine (SVM) with a linear kernel; and (iii) multinomial Bayes (MNB). We used the implementations in scikit-learn[7].

5.1 Dataset, Preprocessing and Evaluation Metrics

We concatenated both the Brown Corpus samples and the extended WOZ dataset (§3) to use as our training and test sets. As the sentences are multiply annotated, we further filtered the annotations from § 3.1 by choosing only the sentences where both annotators agreed, in the case of two annotators, and where three or more agreed in the case of five annotators. This gave a total of 1,337 sentences. We tested our proposed classifiers on the filtered annotated data via 10-fold cross-validation. We report the average accuracy across the different folds.

We tokenised all sentences in the dataset as a preprocessing step. Stop words were only removed for the **word embedding** representation. We did not remove stop words for **tf-idf** because the approach

[5]`https://spacy.io/`. We use the model `en_vectors_web_lg`.
[6]`http://commoncrawl.org/the-data/`
[7]`http://www.scikit-learn.org/`

Model	**VerbNet**	**WordNet**	**tf-idf**	**Embedding**	**WordNet+tf-idf**	**Embedding+tf-idf**
kNN	0.5004	0.5633	0.5370	0.7090	0.6553	0.7868
SVM	0.5355	0.6104	0.7426	0.7771	0.7509	**0.7891**
MNB	0.5370	0.6238	0.7188	–	0.6941	–

Table 2: Accuracy results. Note that the MNB classifier does not support the negative values of the **Embedding**-based representation.

Model	**VerbNet**	**WordNet**	**tf-idf**	**Embedding**	**WordNet+tf-idf**	**Embedding+tf-idf**
kNN	0.2901	0.4543	0.5124	0.6804	0.5624	0.6941
SVM	0.1786	0.5108	0.7380	**0.7434**	0.7324	0.7378
MNB	0.1790	0.4928	0.4647	–	0.4536	–

Table 3: Balanced Accuracy Results.

worked better without stop word removal. We also did not remove stop words for the explicit representations, as they require the complete sentence for parsing to be performed.

We found the filtered dataset to be skewed towards class **0** (non-VDL, 53.70%), compared to classes **1** (fully VDL, 9.65%) and **2** (partially VDL, 36.65%). Thus, besides the *accuracy* metric, we also evaluated our classifiers using a *balanced accuracy* metric to account for the class imbalance. Formally:

$$Acc_{balanced} = \frac{1}{N}\sum_{i=1}^{N}\frac{P_i}{M_i} \tag{1}$$

where N is the number of classes, P_i is the number of correct predictions of class i, and M_i is the number of instances of class i.

5.2 Classification Results

Table 2 shows the accuracies using the three classifiers with 10-fold cross-validation. Note that we did not test the MNB classifier on **Embedding**-based representations as MNB does not allow negative feature values (word embeddings can have negative values). We can see that the **VerbNet** approach with the kNN classifier scored the lowest accuracy, whereas the **WordNet** approach was slightly better. Accuracy scores increased with **tf-idf** equal to 0.7426 with the SVM and the **Embedding** feature scoring 0.7771. In addition, combining **Embedding** with **tf-idf** increased the accuracy, with a score of 0.7891. On the other hand, combining **WordNet** with **tf-idf** only slightly increased the accuracy, giving 0.7509 for SVM.

Table 3 shows the results using our balanced accuracy metric. The balanced accuracy metric in all cases gave lower numbers than standard accuracy. Results significantly dropped for the **VerbNet** approach across all three classifiers, and there is also the substantial drop for **WordNet**. In addition, the results also show a huge drop for the **tf-idf** in the case of the MNB classifier, but not the kNN or SVM classifiers. Results for **tf-idf**, **Embedding**, **WordNet+tf-idf** and **Embedding+tf-idf** with the SVM classifier are comparable, with **Embedding** having a slight edge, giving the highest balanced accuracy score of 0.7434.

5.3 Analysis

Multiple factors affect the accuracy of the classifiers based on **WordNet** and **VerbNet** representations. One is parser inaccuracy, which may lead to incorrect identification of the main verb and/or the subject and object associated with the verb. The VerbNet database may also not cover all English verbs, so some verbs may be unclassified. In addition, VerbNet has many classes of verbs (237 classes) compared to

WordNet (15 classes). This is likely to reduce the accuracy of prediction in the case of Verbnet and may well be the reason why WordNet's accuracy results are higher than those of VerbNet. Section 5.3.1 below provides more in-depth analysis of errors caused by either the parser or the WordNet database.

As we previously reported, class **1** represents only a small part of the data. This class imbalance may also be an issue with learning to classify VDL.

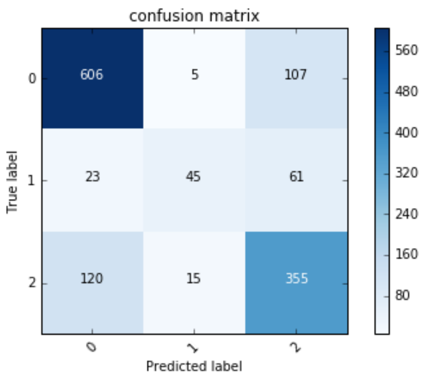

Figure 3: Tf-idf Confusion Matrix

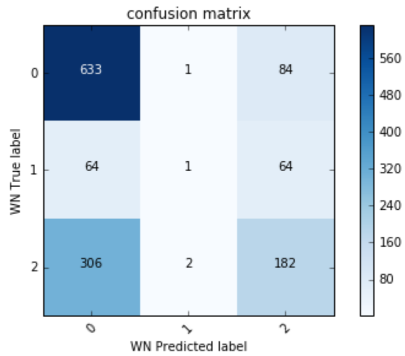

Figure 4: WordNet Confusion Matrix

Figure 3 shows the confusion matrix for the **tf-idf** representation with the SVM classifier. For class **0** (non-VDL), 112 of 718 were misclassified (5 were misclassified as class **1**, and 107 were misclassified as class **2**). For class **1** (fully VDL), only 45 of the 129 total were classified correctly, whereas 61 were misclassified as class **2** and 23 were misclassified as class **0**. For class **2** (partially VDL), 355 out of 490 were correctly classified, whereas 15 were misclassified as class **1** and 120 were misclassified as class **0**. This classifier frequently misclassified text that contains VDL. Figure 4 shows another confusion matrix, in this case for the **WordNet** representation with the MNB classifier.

5.3.1 Error Analysis

We also analysed the types of errors that occurred during the feature extraction stage. We identified various categories of errors made by the parser and also the number of words missing from the WordNet database. The data consists of 1337 sentences, the following shows the percentages of errors found in the data.

1. **Incorrect verbs**: 24.83% of the verbs were incorrect verbs; by incorrect verb here we mean that the word extracted by the parser from the sentence as the main verb was not actually a verb.

2. **Missed subjects**: 20.49% of the subjects were missed by the parser; this meant that no subject was found for the main verb extracted by the parser.

3. **Missed objects**: In 69.93% of the sentences, no object was found for the main verb extracted by the parser. Sentence-level checking has not been carried out to confirm in how many cases an object should have been but it is unlikely that such a large proportion of the sentences contains intransitive verbs.

4. **WordNet subject errors**: 31.79% of the subjects extracted by the parser were not found in the WordNet database. In some cases this was because the subject was a proper nouns – a person name, street name or place name, for example.

5. **WordNet object errors**: 7.26% of the objects extracted by the parser were not found in the WordNet database. As with the subject errors, this may be due to objects being proper nouns.

5.4 Classifying VDL in Other Texts

To demonstrate that our sentence-level VDL classifier can be applied to texts by different authors, we also perform an experiment where we attempt to classify sentences from books from Project Gutenberg[8]. We selected three books for each of the five top authors in Project Gutenberg: Charles Dickens, Arthur Conan Doyle, Mark Twain, William Shakespeare, and Lewis Carroll. We segmented the texts into sentences, and classified all sentences using the SVM sentence-level classifier with the **Embedding+tf-idf** representation as it is the most accurate classifier in our experiments.

Figure 5 shows the distribution of our classifer's predictions of non-VDL (**0**), fully VDL (**1**), and partially VDL (**2**). Conan Doyle and Twain have the highest proportion of pure VDL sentences, while Twain and Dickens have the largest proportion of sentences classified as partially VDL. On the other hand, most of Carroll's and Shakespeare's sentences are not classified as visually descriptive. Overall, the proportion of sentences classified as fully VDL was small, mirroring the class imbalance in the training set.

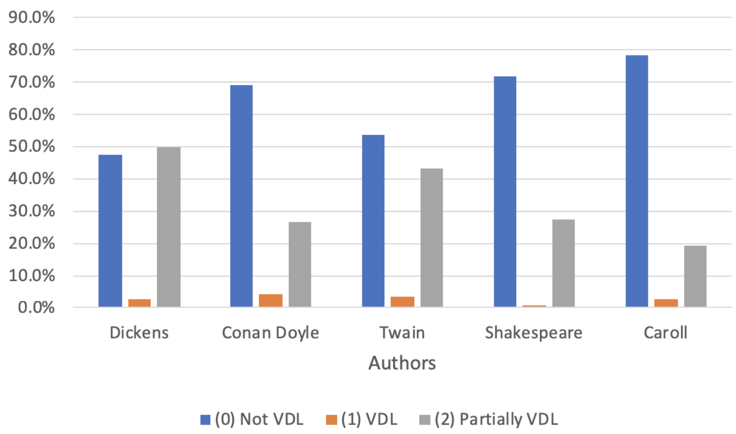

Figure 5: Classification of The Top Five Authors' Books from Project Gutenberg

6 Conclusion

In this paper we have proposed the task of recognising visual language as a potentially interesting and useful challenge for vision and language research. We have recapitulated the definition of VDL as presented in Gaizauskas et al. (2015) and reported the extension of our corpus of VDL-annotated text. We presented initial, promising results of developing classifiers for carrying out the sentence level task of distinguishing sentences that are wholly VDL, partially VDL or not VDL at all. Future work includes devising algorithms for the segment-level annotation task and applying the results of automatic VDL analysis. One application is the extraction and comparison of VDL in various authors' works, some initial results of which we have reported here.

Acknowledgements

Tarfah Alrashid was supported by a PhD studentship from the University of Jeddah. This work was also partially funded by the ERA-Net CHIST-ERA D2K VisualSense project (UK EPSRC EP/K019082/1). The authors thank the two annotators for their time, and also thank the anonymous reviewers for their thorough and useful feedback on an earlier draft of this paper.

[8] `http://www.gutenberg.org/`

References

Antol, S., A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh (2015). VQA: Visual question answering. In *IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, pp. 2425–2433. IEEE.

Bernardi, R., R. Cakici, D. Elliott, A. Erdem, E. Erdem, N. Ikizler-Cinbis, F. Keller, A. Muscat, and B. Plank (2016). Automatic description generation from images: A survey of models, datasets, and evaluation measures. *J. Artif. Intell. Res.(JAIR) 55*, 409–442.

Chattopadhyay, P., D. Yadav, V. Prabhu, A. Chandrasekaran, A. Das, S. Lee, D. Batra, and D. Parikh (2017). Evaluating visual conversational agents via cooperative human-AI games. In *Proceedings of the Fifth AAAI Conference on Human Computation and Crowdsourcing (HCOMP)*.

Chen, D. and C. Manning (2014). A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, pp. 740–750. Association for Computational Linguistics.

Das, A., S. Kottur, K. Gupta, A. Singh, D. Yadav, J. M. F. Moura, D. Parikh, and D. Batra (2017). Visual dialog. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, pp. 1080–1089. IEEE.

Dodge, J., A. Goyal, X. Han, A. Mensch, M. Mitchell, K. Stratos, K. Yamaguchi, Y. Choi, H. Daumé III, A. C. Berg, et al. (2012). Detecting visual text. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 762–772. Association for Computational Linguistics.

Fang, H., S. Gupta, F. Iandola, R. K. Srivastava, L. Deng, P. Dollar, J. Gao, X. He, M. Mitchell, J. C. Platt, C. Lawrence Zitnick, and G. Zweig (2015). From captions to visual concepts and back. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, pp. 1473–1482. IEEE.

Fellbaum, C. (Ed.) (1998). *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press.

Gaizauskas, R., J. Wang, and A. Ramisa (2015). Defining visually descriptive language. In *Proceedings of the Fourth Workshop on Vision and Language*, Lisbon, Portugal, pp. 10–17. Association for Computational Linguistics.

Karpathy, A. and L. Fei-Fei (2015). Deep visual-semantic alignments for generating image descriptions. In *IEEE Conference on Computer Vision & Pattern Recognition (CVPR)*, Boston, MA, USA, pp. 3128–3137. IEEE.

Kulkarni, G., V. Premraj, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg (2011). Baby talk: Understanding and generating simple image descriptions. In *IEEE Conference on Computer Vision & Pattern Recognition (CVPR)*, Colorado Springs, CO, USA, pp. 1601–1608. IEEE.

Kuznetsova, P., V. Ordonez, A. C. Berg, T. L. Berg, and Y. Choi (2012). Collective generation of natural image descriptions. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pp. 359–368. Association for Computational Linguistics.

Manning, C. D., M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky (2014). The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, Baltimore, MD, USA, pp. 55–60. Association for Computational Linguistics.

Pennington, J., R. Socher, and C. Manning (2014). GloVe: global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, pp. 1532–1543. Association for Computational Linguistics.

Schuler, K. K. (2005). *Verbnet: A Broad-coverage, Comprehensive Verb Lexicon*. Ph. D. thesis, University of Pennsylvania, Philadelphia, PA, USA.

Vinyals, O., A. Toshev, S. Bengio, and D. Erhan (2015). Show and tell: A neural image caption generator. In *IEEE Conference on Computer Vision & Pattern Recognition (CVPR)*, Boston, MA, USA, pp. 3156–3164. IEEE.

Yu, L., E. Park, A. C. Berg, and T. L. Berg (2015). Visual Madlibs: Fill in the blank description generation and question answering. In *IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, pp. 2461–2469. IEEE.

Zhu, Y., O. Groth, M. Bernstein, and L. Fei-Fei (2016, June). Visual7W: Grounded question aswering in images. In *IEEE Conference on Computer Vision & Pattern Recognition (CVPR)*, Las Vegas, NV, USA, pp. 4995–5004. IEEE.

Plug-ins for content annotation of dialogue acts

Harry Bunt
Department of Cognitive Science and Artificial Intelligence
Tilburg University, Tilburg, The Netherlands
harry.bunt@uvt.nl

Abstract

Experience in using the concepts of the ISO 24617-2 standard for dialogue act annotation and the DIT++ dialogue act taxonomy has made it clear that a severe limitation for their on-line use in dialogue systems is the lack of semantic content. This paper explores a possibility to remedy this by introducing the concept of an 'annotation plug-in'. It is argued that this can help to make these schemes more directly useful for on-line use as well as to increase both the coverage and the precision of the annotations they support.

1 Introduction

Dialogue act taxonomies, such as those of DIT++ and the ISO 24617-2 annotation standard, have two prime use cases: annotation of dialogue corpora and on-line use in dialogue systems. The DIT++ taxonomy was originally developed as on the one hand an instrument in the analysis of dialogue structure and dialogue mechanisms[1], and on the other hand a basis for the design of dialogue management modules in interactive systems[2]. The theory underlying both schemes, Dynamic Interpretation Theory, defines a dialogue act much in the spirit of speech act theory as having as its most important components a communicative function and a semantic content (besides dependence relations to other dialogue acts and possibly certain qualifiers). As the DIT++ taxonomy evolved into the ISO 24617-2 standard for dialogue act annotation (also referred to as 'ISO-DiAML'), a strong focus came to lie on its use for annotating the functional aspects of dialogue utterance meanings, and thus on communicative functions and inter-dependences of dialogue acts at the expense of attention to their semantic content, since other ISO standards were developing for aspects of semantic content. This limits the on-line use of ISO-DiAML in dialogue systems, where the semantic content of utterances is equally important as their communicative function.

The dialogue acts defined in ISO-DiAML have in recent years been used in dialogue management systems, getting around its limitations by invoking a semantic parser or entity extraction engine, as provided in NLU tools like LUIS or RASA for chatbots,[3] to add semantic content in a way that is suitable for the task domain. This paper explores a more general approach in the form of 'annotation plug-ins', annotation mini-schemes that can be added on to a host annotation scheme and provide semantic content information in various degrees of detail.

This paper is organized as follows. Section 2 very briefly provides some background and motivation for the rest of the paper. Section 3 summarises the definition of the DiAML markup language for dialogue act annotation, defined as part of the ISO-DiAML standard, including its tripartite syntactic and

[1] See e.g. Geertzen and Bunt (2006), Keizer and Morante (2006), Bunt et al. (2007), Geertzen (2009), Petukhova and Bunt (2009a, b), Wlodarczak et al. (2010), Bunt (2011), Fang et al. (2011), Petukhova et al (2011), Bunt (2012), Petukhova et al. (2015), Fang et al. (2018), and related studies such as Besser and Alexandersson (2007), Piwek (1998), Traum and Hinkelman (1992), Traum (1994), Poesio and Traum (1997).

[2] E.g. in the TENDUM system, see Bunt et al. (1984); in the DENK system (Bunt et al. 1998), and in the PARADIME system, see Keizer et al. (2011)

[3] LUIS: https://docs.microsoft.com/en-us/azure/cognitive-services/luis/what-is-luis; RASA: https://www.rasa.com/docs/nlu/

semantic definition. Section 4 introduces the device of annotation plug-in, first for adding very simple, ad hoc annotations of semantic content, and second for adding more principled annotations based on ISO standards. Section 5 considers the use of plug-ins for other purposes, first, for allowing more accurate annotations of discourse relations in dialogue, and second for supporting the import of task-specific types of dialogue act. Section 6 concludes by indicating the perspectives for extending the idea of annotation plug-ins.

2 Background

ISO standards are examined every five years for the need of correction, extension, or other updates. Since ISO-DiAML was formally established late 2012, it has come up for consideration in late 2017.

Experiences in using the ISO-DiAML scheme in annotation projects and dialogue system development have led to the awareness of certain limitations and desirable improvements, discussed in Bunt et al. (2017) and Bunt et al. (2018). These limitations concern mainly (1) the lack of semantic content information; (2) the absence of domain-specific dialogue act types; (3) the somewhat sloppy annotation of rhetorical relations, and (4) the sometimes inaccurate annotation of certain dependence relations.

Of these limitations, the latter one will be addressed in a revision of the standard. The third limitation can only be overcome by somehow adding semantic content information to dialogue act annotations, and the first two limitations are inherent to the scope of ISO-DiAML, and can only be tackled by radically extending its scope, which is not really an option. In order to nonetheless deal with these limitations, we introduce in this paper the notion of a 'plug-in', which allows semantically richer and customised annotations, and enhances the on-line usability in dialogue systems.

3 DiAML

3.1 Annotation scheme architecture

The ISO-DiAML annotation scheme has been designed according to the ISO principles of semantic annotation (ISO standard 24617-6, 'SemAF Principles', see also Bunt (2015) and Pustejovsky et al. (2017). This means that the scheme has a three-part definition consisting of (1) an abstract syntax that specifies the possible *annotation structures* at a conceptual level as set-theoretical constructs, such as pairs and triples of concepts; (2) a semantics that specifies the meaning of the annotation structures defined by the abstract syntax; (3) a concrete syntax, that specifies a representation format for annotation structures (for example using XML).

Defining the semantics at the level of the abstract syntax puts the focus of an annotation standard at the conceptual level, as required by the ISO Linguistic Annotation Framework (ISO 24612, see also Ide and Romary, 2004) . rather than at the level of representation formats. Annotators (human or automatic) deal with concrete representations only, but they can rely on the existence of an underlying abstract syntax and its semantics.

3.2 Abstract syntax

The abstract syntax specifies a store of basic concepts, called the 'conceptual inventory'. The DiAML conceptual inventory consists of:

- a set of nine dimensions (Task, Auto-Feedback, Allo-Feedback, Turn Management, ...);
- a set of communicative functions;
- a set of qualifiers, partitioned into those for certainty, for conditionality, and for sentiment;
- a set of dialogue participants;
- a collection of primary data, segmented into markables.

Given a conceptual inventory, the abstract syntax specifies certain set-theoretical structures like pairs, triples, and more complex nested structures made up from the elements of the inventory. Two types of structure are distinguished: *entity structures* and *link structures*. An entity structure contains semantic information about a segment of primary data and is formally a pair ⟨m, s⟩ consisting of a markable, which refers to a segment of primary data, and certain semantic information. A link structure contains information about the way two or more segments of primary data are semantically related.

A dialogue act is characterised by eight components: (1) a sender S, (2) one or more addressees A, (3) zero or more other participants H, (4) a dimension D, (5) a communicative function F, (6) zero or more dependence relations to a set E of other dialogue acts, (7) zero or more qualifiers Q, and (8) a semantic content c, where the components H, T, and Q are not necessarily present. In DiAML a full-blown entity structure is a septet $\langle S, A, H, D, F, E, Q\rangle$ rather than an octet, since the semantic content is not annotated. For example, the entity structure for a task-related question, expressed through the markable m_1, and addressed by participant p_1 to participant p_2, is the nested quadruple in (1)

(1) $\langle m_1, \langle p_1, \{p_2\}, Task, Question\rangle\rangle$

A link structure in DiAML is a triple $\langle e, E, R\rangle$ consisiting of an entity structure e, a set of entity structures E, and a rhetorical relation R. For example, the rhetorical relation represented in (14) corresponds in the abstract syntax to the link structure $\langle da_2, da_3, Cause\rangle$

A full-blown annotation structure for a dialogue in DiAML is a set $\{e_1, ..., e_n, L_1, ..., L_k\}$ of entity structures structures $(e_1, ..., e_n)$ and link structures $(L_1, ..., L_k)$. For example, the annotation structure for a dialogue fragment consisting of a single question-answer pair, is the following nested structure, in which the entity structure for the question re-appears inside the one for the answer, indicating that the answer is an answer to that question (a 'functional dependence' relation):

(2) $\{\langle m_1, \langle p_1, \{p_2\}, Task, Question\rangle\rangle,$
$\langle m_2, \langle p_2, \{p_1\}, Task, Answer, \{\langle m_1, \langle p_1, \{p_2\}, Task, Question\rangle\rangle\}\rangle\rangle\}$

3.3 Concrete syntax

The conceptual structures defined by the DiAML abstract syntax can be represented in a variety of ways; the ISO 24617-2 standard specifies a pivot XML-based representation format (also referred to as 'DiAML-XML').

For the representation of entity structures an XML element <dialogueAct> is defined, with an attribute @xml:id whose value is a unique identifier; an attribute @target, whose value anchors the annotation in the source data, having a markable (or a sequence of markables) as its value; and furthermore the following attributes, corresponding to its conceptual components: @sender, @addressees, @other participants (optional), @dimension, @communicative function, @dependences (optional), and @qualifiers (optional). For example, the entity structure shown in (1) is represented as:

(3) <dialogueAct xml:id="da1" target="#m1" sender="#p1" addressee="#p2"
dimension="task" communicativeFunction="question"/>

For the representation of link structures the XML element <rhetoricalLink> is defined, with the attributes @dact, @rhetorelatum, and @relType. Their use is illustrated in example (14).

3.4 Semantics

The DiAML semantics consists of the specification of a recursive interpretation function I_{DA} that defines a functor which, applied to a semantic content, forms an information state update operation. The DiAML semantics is compositional in the sense that the interpretation of an annotation structure is obtained by combining the interpretations of its component entity structures and link structures; see Bunt ((2014) for details.

Formally, according to SemAF Principles (ISO 24617-6) an annotation scheme is defined by a triple formed by specifications of an abstract syntax (AS), a concrete syntax (CS), and a semantics (SEM), each of which is further structured:

(4) $A = \langle AS, CS, \text{SEM}\rangle = \langle\langle CI, AC\rangle, \langle VC, CC, eF\rangle, \langle M, I\rangle\rangle$

The abstract syntax specification consists of the conceptual inventory CI and the specification of conceptual structures AC; together, these define the class of well-formed annotation structures.

The concrete syntax specification CS contains a vocabulary VC, the specification CC of a class of syntactic structures, such as XML elements, and an encoding function (including a mapping from AC to CC). The components VC and CC together define a class of well-formed representations, and eF assigns such a representation to every well-formed annotation structure.

The semantic specification SEM is in general a pair $\langle M, I\rangle$, consisting of a model and an interpretation function. For the ISO-DiAML host annotation scheme, the semantics uses a context model (or 'information state') for M and an interpretation function defined in terms of context updates.

4 Content plug-ins

The semantic content of a dialogue act, expressed by a markable (functional segment) m_i, can of course be supplied by a semantic parser-interpreter that delivers the semantic content of m_i. This is what Keizer et al (2011) and Malchanau (2019) do for an application domain where the semantic content has a simple structure, that can be described by a list of attribute-value pairs. Here we explore an alternative possibility.

The various parts of the ISO Semantic Annotation Framework (SemAF, ISO 24617) other than ISO-DiAML are all concerned with the annotation of aspects of sentence meaning, and these annotations have a compositional semantics. A methodologically elegant option that presents itself would be to combine ISO-DiAML with some of these other annotations and use their semantics to obtain the semantic content of dialogue acts. This can be accomplished by defining 'plug-ins' for the DiAML annotation language, as described in the rest of this section.

4.1 A domain-specific content plug-in

To introduce the idea of an annotation plug-in, we first consider the case of a simple domain-specific plug-in that could for example be useful in a journey planning domain where a task can be described by a few attribute-value pairs, specifying departure place, destination, travel date, etc. For example, the utterance *"I would like to leave around ten in the morning"* from a client PA1 could be semantically annotated as in (5b):

(5) a. PA1: "I would like to leave around ten in the morning" (= markable m1)

b. <avContent xml:id="av1" target="#m1" attribute="departureTime" value="10:00"/>

According to the ISO principles of semantic annotation, as laid down in ISO 24617-2 (see also Bunt, 2015), a semantic annotation should have an underlying abstract syntax and a semantics. Underlying the representation used in (5b) would be a conceptual inventory that lists the attributes and their possible values, and an entity structure in the form of a nested pair $\langle m, \langle A_i, v_{ij}\rangle\rangle$ made up of a markable, an attribute and a value (taken from the conceptual inventory). The semantics of such an entity structure could e.g. be a feature structure of the form $[A'_i : v'_{ij}]$ which, according to the ISO standard 24612 for feature structures can be viewed as the function $\lambda x.A'_i(x) = v'_{ij}$.

The syntax and semantics of such AV-entity structures define a very simple annotation language L_{AV}, the semantics of which is a defined by:

(6) $I_{AV}(\langle A_i, v_{ij}\rangle) = [I_{AV}(A_i) : I_{AV}(v_{ij})] = [A'_i : v'_{ij}]$

Attribute-value pairs $\langle A_i, v_{ij}\rangle$, their XML encoding as in (5b), and the specification of their semantics as in (6) in fact define a mini-annotation-scheme for semantic content annotation that can be combined with DiAML annotation – such a mini-scheme is what we call an annotation *plug-in*. A plug-in is thus formally characterised as a triple $PL_a = \langle AS_a, CS_a, \text{SEM}_a\rangle$. The formal specification of the attribute-value content plug-in PL_{AV} for DiAML is as follows:

- AS_{AV}: the conceptual inventory lists attributes and their possible values; entity structures for semantic content of the form ⟨markable, ⟨attribute, value⟩⟩.
- CS_{AV}: vocabulary items for attributes and values; encoding of entity structures as in (5b).
- SEM_{AV}: I_{AV} as in (6).

To integrate plug-in annotations of semantic content with DiAML-annotations of the functional aspects of dialogue acts, there are two options.

- extend the entity structures describing a dialogue act from septets to octets, including their semantic content, and add a corresponding attribute @semContent to the XML encoding of in a <dialogueAct> element. This leads to representations of the following form:

  (7) <dialogueAct xml:id="da1"target="#m1" speaker="#s" addressee="#a"
  dimension="task" communicativeFunction="inform" semContent="#av1"/>
  <avContent xml:id="av1" target="#m1" attribute="departureTime" value="10:00"/>

- introduce a new link structure that relates a dialogue act to its semantic content, and add a corresponding <contentLink> element in the XML encoding. This leads to representations in the following form:

  (8) <dialogueAct xml:id="da1"target="#m1" speaker="#s" addressee="#a"
  dimension="task" communicativeFunction="inform"/>
  <avContent xml:id="av1" target="#m1" attribute="departureTime" value="10:00"/>
  <contentLink dAct="#da1" content="#av1"/>

According to the underlying theory, a dialogue act is formally characterized as an octet, including a semantic content (see Section 3.2 above), so the first option may seem most plausible. However, it requires the introduction in DiAML of an attribute like @semContent of which the possible values cannot be specified, since that depends on the plug-in that is used. The second option has several interesting advantages:

1. it makes it very clear that the dialogue annotation of a dialogue act does not require the specification of a semantic content: the use of the plug-in is optional.[4]

2. the use of an explicit link between the functional aspects of a dialogue act and its semantic content allows the specification of additional information attached to the link, such as (un-)certainty scores and alternatives, supporting the management of semantic ambiguity.

3. the annotation of dialogue acts in DiAML remains the same, therefore any formal operations and any software defined for DiAML annotations are still applicable.

We therefore choose the second option. The introduction of a content link structure shows that the combination of a host annotation scheme with a plug-in requires in general an *interface* that connects the two. The content link structure is neither part of DiAML nor of the plug-in, but is part of the interface between the two. An interface ${}^{p}Y_h$ between a host scheme h and a plug-in p can be formally specified in a similar way as the host scheme itself (see (4)) and the plug-in, with an abstract and concrete syntax

[4]See Bunt (2018) for a discussion of different forms of optionality in semantic annotations.

and a semantics, except that the interface does not require the introduction of new basic concepts or vocabulary items, but only of one or more link structures for relating host and plug annotations.

For the present plug-in, the abstract syntax of the interface ${}^{AV}Y_{DA}$ defines the content link structure as a pair $\langle \epsilon_a, \epsilon_c \rangle$ consisting of a dialogue act entity structure and a content entity structure; no relation between the two needs to be specified since no other relation between the two kinds of structure are envisaged. The concrete syntax of the interface defines the <contentLink> element and the encoding of pairs $\langle \epsilon_a, \epsilon_c \rangle$ using this element. As noted above (Section 3.4), the DiAML semantics specifies an interpretation function I_{DA} which defines a functor that, applied to a semantic content, forms an information state update operation. The semantic component of the interface specifies the interpretation of the new content link structure in a way that expresses exactly that:

(9) $I_{{}^{AV}Y_{DA}}(\langle \epsilon_a, \epsilon_c \rangle) = I_{DA}(\epsilon_a)(I_{AV}(\epsilon_c))$

Using DiAML with the PL_{AV} plug-in comes down to applying an extended annotation scheme of which the abstract syntax is formed by the DiAML abstract syntax, the PL_{AV} abstract syntax, and the content link stsructure of the interface; the concrete syntax likewise is that of DiAML extended with PL_{AV}-representations and the <contentLink> element; the semantics is that of DiAML extended with $I_{AV}(\epsilon_{AV})$ for content entity structures and with (9) for content link structures.

A caveat: the semantic part of the interface, as formed by (9), assumes that the interpretation function I_{DA} of the host language is applicable to the output of the plug-in interpretation function I_{AV}. The DiAML semantics makes use of elementary context update operators which are defined in a representation-neutral way, just stipulating for example that a given semantic content should be added to that part of the addressee's information state which contains information about the task that still has to be verified for consistency with other available information (the addressee's 'pending semantic context'). To apply this approach in a dialogue system, the elementary update operators must be instantiated for the representation formalism of the system's information state. The semantic content of dialogue acts has to be represented in a form that fits in with that formalism, and if necessary has to be converted to that. For content expressed in the form of feature structures, as is the case for I_{AV}, this will not be an obstacle. Existing DiAML implementations in dialogue systems, such as Keizer et al. (2011), Malchanau et al. (2017), and Malchanau (2019) use typed feature structures for information representation, making the implementation of (9) a straightforward matter.

It may be noted that the annotation of a semantic content formed by an attribute-value pair, by means of an XML-element <avContent>, can be viewed as just a conveniently compact abbreviated notation of an ISO/TEI-conformant XML-representation using the TEI vocabulary. For example, the representation used in (5b) is equivalent to the XML-representation in (10), which makes use of TEI-defined vocabulary items and of feature structures as specified in ISO standard 24612. The abbreviating representation used in (5), (7), and (8) could be automatically expanded into this representation.

(10)
```
<u xml:id=”m1”>I would like to leave around ten in the morning</u>
  <annotationBlock type=”semanticContent”>
    <spanGrp type=”markable”>
      <span target=”#m1” ana=”#fs1”/ >
    </spanGrp>
  <fs type=”avContent” xml:id=”fs1”>
    <f name=”departureTime”>
      <string>10:00</string>
    </f>
  </fs>
  </annotationBlock>
```

4.2 Semantic roles

We next consider a more general content-plug, based on an ISO standard.

Semantic annotation standard ISO 24617-4 for semantic role labelling, a.k.a. 'SemAF-SR', marks up semantic information related to the question *Who did what to whom?*, assigning semantic roles to the participants in an event. For instance, the example sentence *"I would like to leave around ten in the morning"* would be analysed as mentioning two eventualities, a like-state and a travel-event, and would be annotated as follows, where "like.01" and "leave.01" correspond to verb senses in VerbNet:

(11) a. PA1: "I would like to leave around ten in the morning"
Markables: m1="I", m2="like", m3="leave", m4="ten in the morning"

b. <eventuality xml:id="e1" target="#m2" eventFrame="like.01" eventualityType="state"/>
<entity xml:id="x1" target="#m1" pred="#pa1"/>
<srLink event="#e1" participant="#x1" semRole="experiencer"/>
<eventuality xml:id="e2" target="#m3" eventFrame="leave.01" eventualityType="activity"/>
<srLink event="#e1" participant="#e2" semRole="theme"/>
<srLink event="#e2" participant="#x1" semRole="agent"/>
<entity xml:id="x2" target="#m4" pred="10:00"/>
<srLink event="#e2" participant="#x2" semRole="time"/>

SemAF-SR interprets such annotations as expressing the existence (or denied existence, in case of a negated clause) of certain states or events and participants in certain roles. For the example in (11) the semantics can be expressed by the following DRS (where pa1 is a constant referring to the speaker of the utterance in (11a)):

(12) [e1 e2 x1 x2 | like01(e1), leave01(e2), x1=pa1, x2=10:00, experiencer(e1,x1), theme(e1,e2), agent(e2,x1), time(e2,x2)]

A content plug-in for DIAML consists in this case of the abstract and concrete syntax of the SemAF-SR markup language and the semantic interpretation function (not spelled out in the ISO standard) which produces DRSs like those in (12). The abstract syntax has a conceptual inventory that lists semantic roles and verb senses by reference to VerbNet, defines entity structures for eventualities and their participants, and link structures for relating participants to eventualities in a certain role. The concrete syntax defines XML encodings of the annotation structures defined by the abstract syntax, as illustrated in (11).

When defining a content plug-in for information about semantic roles, the question arises whether *all* the information encoded in SemAF-SR annotations should be taken along in the plug-in. This issue concerns especially the reference to event frames for VerbNet verb senses. While this seems appropriate for the purposes of SemAF-SR, it would bring a level of precision to the interpretation of verbs and deverbal nouns which is not pursued for other content words; it may therefore be more appropriate to make this optional in a plug-in, allowing users to choose whether they want to plug in a conceptual inventory with that level of granularity or a less fine-grained one. ISO-TimeML (ISO 24617-1), the ISO standard for annotating time and events, should also be considered here. It uses a classification of event types that differs from that of SemAF-SR, and it includes other detailed information about events that is not considered in SemAF-SR. Again, it is not obvious how much of that information would seem appropriate to take along in a plug-in for DiAML. A closely related issue concerns the analysis of the semantic roles of temporal objects. SemAF-SR distinguishes three temporal roles: Duration, Initial-time, and Final-time, whereas ISO-TimeML has a much larger and more fine-grained set of temporal relations.

The simplest content plug-in for semantic roles takes a minimalist approach to event classifications, and uses a simple form like <eventuality xml:id="e2" target="#m3" pred="leave"/> rather than the more fine-grained annotations of SemAF-SR or ISO-TimeML. This plug-in ('PL_{SR}') is characterized by the following schema:

Abstract syntax: the conceptual inventory lists the semantic roles defined in ISO 24617-4 and a set of verb senses, distinguishing only between senses which differ in the semantic roles that they take; two kinds of entity structures are distinguished, for eventualities and their participants (other than eventualities), and one kind of link structure, for indicating a semantic role.

Concrete syntax: specifies names for the elements of the conceptual inventory, and defines XML elements for encoding the entity and link structures.

Semantics: translation of entity and link structures and their combination to DRSs.

Content annotation structures according to this plug-in can be linked to dialogue acts using content link structures in the same way as for the AV-plug-in. This allows the example utterance (17a) to be annotated as follows:

(13) a. PA1: ”I would like to leave around ten in the morning”
Markables: m1=“I”, m2=“like”, m3=“leave”, m4=“ten in the morning”

b.
```
<dialogueAct xml:id=”da1” target=”#m1” speaker=”#pa1” addressee=”#a”
   dimension=”task” communicativeFunction=”inform”/>
<eventuality xml:id=”e1” target=”#m2” pred=”like”/>
<entity xml:id=”x1” target=”#m1” pred=”#pa1”/>
<srLink event=”#e1” participant=”#x1” semRole=”experiencer”/>
<eventuality xml:id=”e2” target=”#m3” eventFrame=”leave”/>
<srLink event=”#e1” participant=”#e2” semRole=”theme”/>
<srLink event=”#e2” participant=”#x1” semRole=”agent”/>
<entity xml:id=”x2” target=”#m4” pred=”#10:00”/>
<srLink event=”#e2” participant=”#x2” semRole=”time”/>
<contentLink dAct=”#da1” content=”#e1”/>
```

The interface for this plug-in can be the same as for the AV plug-in, with the semantic part like (9) except that the interpretation function I_{SR} is used instead of I_{AV}.

5 Other plug-ins

5.1 More fine-grained rhetorical relations

ISO 24617-2 supports the marking up of rhetorical relations between dialogue acts, but does not specify any particular set of relations to be used; it only specifies *how* a rhetorical relation may be marked up as relating two dialogue acts. ISO standard 24617-8 for annotating semantic relations in discourse, also known as DR-Core, was established in 2016 and defines a core set of rhetorical relations. Comparing the ISO-DiAML and DR-Core annotation schemes, two limitations have been noted of the way rhetorical relations can be marked up in DiAML. First, many rhetorical relations have two arguments that play different roles, for example, a Cause relation has one argument that plays the role of a reason and another that plays the role of a result. ISO-DiAML has a provision for indicating the existence of a causal relation between two dialogue acts, but not for indicating their roles, as illustrated in (14):

(14) A: Have you seen Pete today?
B: Pete didn’t come in. He has the flu.

```
<dialogueAct xml:id=”da1” target=”#fs1” sender=”#a” addressee=”#b”
   dimension=”task” communicativeFunction=”propositionalQuestion”/>
<dialogueAct xml:id=”da2” target=”#fs2” sender=”#b” addressee=”#a” dimension=”task”
   communicativeFunction=”answer” functionalDependence=”#da1”/>
<dialogueAct xml:id=”da3” target=”#fs3” sender=”#b” addressee=”#a”
   dimension=”task” communicativeFunction=”inform”/>
<rhetoricalLink dact=”#da3” rhetoAntecedent=”#da2” rhetoRel=”cause”/>
```

By contrast, DR-Core annotations make argument roles explicit, as illustrated in the DR-Core annotation of B's two utterances in (15):

(15) A: Have you seen Pete today?
B: Pete didn't come in. He sent me a message saying that he has the flu.
<drArg xml:id="a1" target="#m1" type="dialogAct"/>
<dRel xml:id="r1" rel="cause"/>
<drArg xml:id="a2" target="#m2" type="event"/>
<drLink rel="#r1" arg1="#a1" arg1Role="result" arg2="#a2" arg2Role="reason"/>

Second, many rhetorical relations may occur either between between the semantic contents of two dialogue acts, or between the semantic content of one dialogue act and the performance of another. This phenomenon is known in the literature as the 'semantic-pragmatic' distinction, and is illustrated by the difference between (14) and (15). B's two utterances in (14) are causally related in the sense that the semantic content of the second dialogue act forms the reason why the content of the first dialogue act is true. In (15), by contrast, there is a 'pragmatic' causal relation, in the sense that the second utterance expresses why B performs the dialogue act of informing A that Pete is not in. This distinction is represented in DR-Core by indicating the types of the arguments, where "dialogAct" is one of the possible types, and the possible types of the semantic content of a dialogue act are the other. This distinction cannot be expressed in DiAML.

This can be remedied in the presence of a plug-in for semantic content annotation, in which case the necessary entity structures are already available, by adding a plug-in which provides a link structure, corresponding to the 'drLink' structure of DR-Core. for annotating the occurrence of a rhetorical relation. A simple plug-in PL_{DR} for rhetorical relations can then be defined as follows:

Abstract syntax:

- Conceptual inventory: the discourse relations defined in DR-Core;
- Link structures: those of DR-Core for rhetorical relations between entity structures as defined in DiAML or in the content plug-in.

Concrete syntax: XML names for the relations in the conceptual inventory and for their argument roles (like '@reason' and '@result'); encodings of DR-Core link structures.

Semantics: The DR-Core interpretation of discourse relations as binary predicates.

For the semantic interpretation of a rhetorical link between a dialogue act and the semantic content of another one, or between the semantic contents of two dialogue acts, the following semantic interface is needed:

(16) $I_{SRY_{DA}}(\langle\epsilon_1, \epsilon_2, \rho\rangle) = I_{DR}(\rho)(\epsilon'_1, \epsilon'_2)$, where ϵ'_i is the representation of ϵ_i in the DiAML context model if ϵ is a DiAML (dialogue act) entity structure; and if ϵ_i is a content entity structure then ϵ'_i is the representation in the context model of the semantic content plug-in.[5]

It may be noted that DR-Core is limited to annotating strictly semantic discourse relations with only a small set of 18 core relations. This is too limited for many applications, as has for example been noted when adding rhetorical relations in the DialogBank (Bunt et al., 2018). A more powerful plug-in could have a two-part conceptual inventory where one part corresponds to the DR-Core set of relations and another part to additional relations, needed for a given annotation task. Moreover, if one wants to annotate additional aspects of discourse relations such as their argument order in the discourse, as in the CCR theory of discourse (Sanders et al., 1992, 2018), then this could be taken care of by introducing relation qualifiers in the plug-in, in a similar way as dialogue act qualifiers are used in DiAML.

[5]The DIAML semantics assumes its context model to include a Dialogue History, a chronological representation of the dialogue acts that constitute a dialogue and of the relations between them (see Bunt, 2014). Clause (16) inserts such relations in the context model, taking semantic content into account.

5.2 Application-specific dialogue act types

The DIT++ dialogue act taxonomy and the ISO-DiAML annotation scheme were both designed to be domain- and task-independent, i.e. to be applicable in virtually every task domain. This is part of their strength, but it is also a limitation, since tasks other than the exchange of information may involve task-specific dialogue act types. For example, the chair person of a meeting may perform meeting-specific dialogue acts such as opening and closing the meeting, and suspending and resuming the meeting; in an interview in a human resource management context, acts such as appointing, promoting, hiring and sacking may occur. For applying ISO-DiAML or DIT++ to meetings and HRM interviews, such communicative functions should be added to the inventories of these annotation schemes.

For negotiation dialogues, Petukhova et al. defined 15 dialogue act types, such as Elicit-offer-value (*“How do you feel about...?”*), Offer-value (*“I could live with just a ban in public transportation”*), Counter-offer-value (*“I go for twenty five then if you’re so bad”*), Bargain-down (*“Okay, I can go for somewhat less restrictive’”*), Deal (*“That’s a deal!*), and Exit-deal (*“We have to re-discuss this.”*). For each of these a context-update semantics is defined[6] in similar terms as for the ISO-DiAML dialogue act types, making their addition conceptually relatively easy. A plug-in would just list the 15 negotiation-specific functions as the conceptual inventory; no additions to the DiAML entity structures or link structures are needed, and hence no additions to the concrete XML representations other than the names of the additional 15 functions. Semantic accommodation is very simple since the semantic component of the plug-in consists of the context-update specifications of the 15 additional functions, which do not interfere with the DiAML semantics and can simply be added.

5.3 Emotions

ISO 24617-2 has no provisions for expressing the emotional aspect that a dialogue act may have. The W3C recommendation EmotionML[7] was designed in part with the aim to serve as a plug-in for other annotation schemes. It characterizes emotions as complex entities, including an emotion category such as *irritated, excited*, or *amused*, an intensity (‘valence’), and , and various alternative ways of characterizing emotions, notably in terms of ‘action tendencies’, ‘appraisals’, and multiple ‘dimensions’. Since EmotionML is defined only at the level of concrete syntax, it cannot directly be used as a plug-in for semantic annotation, however, the concrete syntax could be used as the starting point for applying the CASCADES development method (see Pustejovsky et al., 2017) to build a full-fledged plug-in.

An EmotionML-based plug-in could use an <emoLink> element as the interace in the concrete syntax for relating the sender of a dialogue act to an EmotionML annotation representation, with corresponding abstract link structures and semantics in terms of updates of the senders emotional state. The following example illustrates the possibilities.

(17) a. PA1: Would you like to have a cup of coffee?
PA2: That would be wonderful!

b. <dialogueAct xml:id=“da1” target=“#m1” speaker=“#pa1” addressee=“#pa2”
dimension=“social” communicativeFunction=“offer”/>
<dialogueAct xml:id=“da2” target=‘#m2” speaker=‘#pa2” addressee=‘#pa1”
dimension=“social” communicativeFunction=‘acceptOffer” funcDep=“#da1”/>
<event xml:id=“e1” target=“#m2” pred=“have-coffee”/>
<srLink event=“#e1” participant=“#pa2” semRole=‘agent”/>
<contentLink dAct=“#da1” content=“#e1”/>
<emotion xml:id=“em1” target=“#m2” category=“happy” valence=“0.8”/>
<emoLink dialogAct=“#da2” emotion=“#em1”/>

[6] See EU project Metalogue Deliverable 4.1, Annex 11.3.
[7] Available at http://www.w3.org/TR/2014/REC-emotionml-20140522/.

6 Conclusions and perspectives

In this paper we have introduced the concept of a plug-in for annotation schemes, and shown how this device opens possibilities for overcoming certain limitations of the ISO-DiAML and DIT++ annotation schemes. As a matter of principle, the ISO-DiAML standard does not deal with the semantic content of dialogue acts and includes only domain-independent communicative functions. Both these restrictions limit the possibility of using the standard on-line in dialogue systems without defining domain-related extensions. Plug-ins for semantic content and for domain-specific functions may form a well-defined and flexible way to overcome these limitations. Such plug-ins have been implemented in DIT++ release 5.2 (see `https://dit.uvt.nl`).

A plug-in for more precise annotation of rhetorical relations in dialogue was described, based on the ISO DR-Core standard, which presupposes a plug-in for semantic content annotation. The support of multiple plug-ins can clearly make the dialogue annotation scheme not only more powerful but also more accurate. Plug-ins derived from ISO-TimeML and ISO-Space (ISO 24617-7) could add temporal and locational information to the events introduced by the semantic roles plug-in.

Two crucial aspects of semantic content annotation that are still missing concern coreference and quantified predicate arguments. For each of these, the development of an ISO annotation standard has recently started (ISO WD 24617-9; ISO WD 24617-12), which opens the perspective of future additional plug-ins to further enhance the ISO-DiAML scheme for dialogue act annotation, analysis and interpretation, making it more useful both for annotation and for on-line use in intelligent interactive systems.

References

Besser, J. and J. Alexandersson (2007). A comprehensive disfluency model for multi-party interaction. In *Proceedings 8th SIGDIAL Workshop on Discourse and Dialogue*, Antwerp, pp. 182 – 189.

Bunt, H. (2011). Multifunctionality in dialogue. *Computer, Speech and Language 25*, 222–245.

Bunt, H. (2012). The semantics of feedback. In *Proceedings of SeineDial, 16th Workshop on the Semantics and Pragmatics of Dialogue*, Paris.

Bunt, H. (2014). A context-change semantics for dialogue acts. In H. Bunt, J. Bos, and S. Pulman (Eds.), *Computing Meaning, Vol. 4*, pp. 177–201. Berlin: Springer.

Bunt, H. (2015). On the principles of semantic annotation. In *Proceedings 11th Joint ACL-ISO Workshop on Interoperable Semantic Annotation (ISA-11)*, London, pp. 1–13.

Bunt, H., R. Ahn, R. Beun, T. Borghuis, and C. van Overveld (1998). Multimodal cooperation with the DENK system. In H. Bunt, R.-J. Beun, and T. Borghuis (Eds.), *Multimodal Human-Computer Communication*, pp. 39–67. Berlin: Springer.

Bunt, H., R. Beun, F. Dols, J. van der Linden, and G. Schwartzenberg (1984). The theoretical foundations of the TENDUM system. *IPO Annual Progress Report 19*, 105–113.

Bunt, H., E. Gilmartin, C. Pelachaud, V. Petukhova, L. Prévot, and M. Theune (2018). Downward compatible revision of the dialogue act annotation. In *Proceedings 14th Joint ISO - ACL Workshop on Interoperable Semantic Annotation (ISA-14)*, Santa Fé, New Mexico, pp. 21–34.

Bunt, H., R. Morante, and S. Keizer (2007). An empirically based computational model of grounding in dialogue. In *Proceedings of the SIGdial Workshop on Discourse and Dialogue*, Antwerp, Belgium, pp. 283–290.

Bunt, H., V. Petukhova, and A. Fang (2017). Revisiting the ISO Standard for Dialogue Act Annotation. In *Proceedings 13th Joint ISO - ACL Workshop on Interoperable Semantic Annotation (ISA-13)*, Montpellier, pp. 37–50.

Bunt, H., V. Petukhova, A. Malchanau, and K. Wijnhoven (2018). The DialogBank: Dialogues with interoperable annotations. *Language Resources and Evaluation 2018*. url: https://doi.org/10.1007/s10579-018-9436-9.

Bunt, H., J. Pustejovsky, and K. Lee (2018). Towards an ISO Standard for the Annotation of Quantification. In *Proceedings 11th International Conference on Language Resources and Evaluation (LREC 2018)*, Myazaki, Japan. ELRA.

Fang, A., H. Bunt, J. Cao, and X. Liu (2011). Relating the semantics of dialogue acts to linguistic properties: A machine learning perspective through lexical cues. In *Proceedings of the 5th International Conference on Semantic Computing (IEEE-ICSC 2011*, Stanford University, Palo Alto, California, USA.

Fang, A., Y. Liu, J. Cao, and H. Bunt (2018). Chinese Multimodal Resources for Discourse Act Analysis. In C.-R. Huang, Z. Jing-Schmidt, and B. Mesterernst (Eds.), *The Routledge Handbook of Chinese Applied Linguistics*. Oxford, UK: Routledge.

Geertzen, J. (2009). *The automatic recognition and prediction of dialogue acts. Ph.D. Thesis*. Tilburg University.

Geertzen, J. and H. Bunt (2006). Measuring annotator agreement in a complex, hierarchical dialogue act schema. In *Proceedings SIGDIAL 2006*, Sydney.

Ide, N. and L. Romary (2004). International standard for a linguistic annotation framework. *Natural Language Engineering 10*, 221–225.

ISO (2012a). *ISO 24617-1: 2012, Language Resource Management - Semantic Annotation Framework (SemAF) - Part 1: Time and events*. Geneva: International Organisation for Standardisation ISO.

ISO (2012b). *ISO 24617-2:2012, Language Resource Management - Semantic Annotation Framework (SemAF) - Part 2: Dialogue acts*. Geneva: International Organisation for Standardisation ISO.

ISO (2014a). *ISO 24617-4: 2014, Language Resource Management - Semantic Annotation Framework (SemAF) - Part 4: Semantic roles*. Geneva: International Organisation for Standardisation ISO.

ISO (2014b). *ISO 24617-7: 2014, Language Resource Management - Semantic Annotation Framework (SemAF) - Part 7: Spatial information*. Geneva: International Organisation for Standardisation ISO.

ISO (2015). *ISO 24617-6:2015, Language Resource Management - Semantic Annotation Framework (SemAF) - Part 6: Principles of semantic annotation*. Geneva: International Organisation for Standardisation ISO.

ISO (2016). *ISO 24617-8:2016, Language Resource Management - Semantic Annotation Framework (SemAF) - Part 8: Semantic relations in discourse, Core annotation scheme (DR-Core)*. Geneva: International Organisation for Standardisation ISO.

ISO (2019a). *ISO 24617-12:2019, Language Resource Management: Semantic Annotation Framework (SemAF) - Part 12: Quantification*. Geneva.: International Standard. International Organisation for Standardisation ISO.

ISO (2019b). *ISO 24617-9:2019, Language Resource Management: Semantic Annotation Framework (SemAF) - Part 9: Reference Annotation Framework*. Geneva.: International Standard. International Organisation for Standardisation ISO.

Keizer, S., H. Bunt, and V. Petukhova (2011). Multidimensional dialogue management. In A. van den Bosch and G. Bouma (Eds.), *Interactive Multimodal Question Answering*, pp. 57–86. Berlin: Springer.

Keizer, S. and R. Morante (2006). Dialogue acts as context operators constraining the generation of dialogical discourse. In C. Sidner, J. Harpur, and P. Kühnlein (Eds.), *Proceedings of the Workshop on Constraints in Discourse*, Maynooth, Ireland, pp. 117–124.

Lee, K., L. Burnard, L. Romary, S. Bauman, H. Bunt, E. de la Clergerie, T. Erjavec, T. Declerck, L. Clement, A. Roussanaly, and C. Roux (2004). Towards an International Standard for the Representation of Feature Structures. In *Proceedings 4th International Conference on Language Resources and Evaluation (LREC 2004)*, Paris. ELRA.

Malchanau, A. (2019). *Cognitive Architecture for Multimodal Multidimensional Dialogue Management*. Saarbrücken: PhD Thesis, University of Saarland.

Malchanau, A., V. Petukhova, and H. Bunt (2017). Virtual Debate Coach design: Assessing multimodal argumentation performance. In *Proceedings International Conference on Multimodal Interaction (ICMI 2017)*, Glasgow.

Petukhova, V. and H. Bunt (2009a). Grounding by nodding. In *Proceedings GESPIN, Conference on Gestures and Speech in Interaction*, Poznán.

Petukhova, V. and H. Bunt (2009b). Who's next? Speaker-selection mechanisms in multi-party dialogue. In *Proceedings DiaHolmia, 2009 Workshop on the Semantics and Pragmatics of Dialogue*, Stockholm.

Petukhova, V., A. Malchanau, and H. Bunt (2015). Modelling argumentation in parliamentary debates. In *Proc. of CMNA XV, 15th Workshop on Computational Models of Natural Argument*, Bartinoro, Italy.

Petukhova, V., A. Malchanau, and H. Bunt (2017). Computing negotiation update semantics in multi-issue bargaining dialogues. In *Proceedings 21st Workshop on the Semantics and Prgatics of Dialogue (SEMDIAL 2017 - Saardial)*, Saarbrücken.

Petukhova, V., L. Prévot, and H. Bunt (2011). Multi-level discourse relations between dialogue units. In *Proceedings 6th Joint ACL-ISO workshop on Interoperable Semantic Annotation (ISA-6)*, Oxford, pp. 18–28.

Piwek, P. (1998). *The logic of conversation*. PhD thesis, Tilburg University.

Poesio, M. and D. Traum (1997). Conversational actions and discourse situations. *Computational Intelligence 13(3)*, 309 – 347.

Pustejovsky, J., H. Bunt, and A. Zaenen (2017). Designing annotation schemes: From theory to model. In N. Ide and J. Pustejovsky (Eds.), *Handbook of Linguistic Annotation*, pp. 21–72. Berlin: Springer.

Sanders, T., V. Demberg, J. Hoek, M. Scholman, F. T. Asr, S. Zufferey, and J. Evers-Vermeul (2018). Unifying dimensions in coherence relations: How various annotation frameworks are related. *Corpus Linguistics and Linguistic Theory*. url: https//doi.org/10.1515/clit-2016-0078.

Sanders, T., W. Spooren, and L. Noordman (1992). Toward a taxonomy of discourse relations. *Discourse Processes 15(1)*, 1–35.

Traum, D. (1994). *A computational theory of grounding in natural language conversation.* PhD Thesis, University of Rochester.

Traum, D. and E. Hinkelman (1992). Conversation acts in task-oriented spoken dialogue. *Computational Intelligence, 8(3)*, 575–599.

Wlodarczak, M., H. Bunt, and V. Petukhova (2010). Entailed feedback: evidence from a ranking experiment. In *Proceedings 14th Workshop on the Semantics and Pragmatics of Dialogue (PozDial)*, Poznan, Poland, pp. 159–162.

The Semantics of ISO-Space

James Pustejovsky
Brandeis University
jamesp@brandeis.edu

Kiyong Lee
Korea University
ikiyong@gmail.com

Harry Bunt
Tilburg University
Harry.bunt@uvt.nl

Abstract

An understanding of spatial information in natural language is necessary for many computational linguistics and artificial intelligence applications. In this paper, we outline the basic semantic structure for ISO-Space, an annotation scheme for the markup of spatial relations, both static and dynamic, as expressed in text and other media. We outline the basic formal semantic requirements of a model for spatial information, as expressed in the metamodel for ISO-Space, and demonstrate some illustrative compositions using type-theoretic derivations. We then show how the concrete syntax of the annotation structure for ISO-Space is consistent with the semantics provided for the metamodel.

1 Introduction

The specification for ISO (2019)[1] distinguishes four major types of spatially relevant elements for markup in natural language:

(1) a. SPATIAL ENTITIES: natural or artificial locations in the world that include places, paths, and trajectories (event paths), as well as objects participating in spatial relations.
b. SPATIAL SIGNALS AND SPATIAL MEASURES: linguistic markers that establish relations between places and spatial entities.
c. SPATIAL RELATIONSHIPS: The specific qualitative configurational, orientational, and metric relations between objects.
d. EVENTS AND MOTIONS: Eventualities involving movement from one location to another.

The corresponding metamodel for these elements is represented in Figure 1 below (Lee et al., 2018).

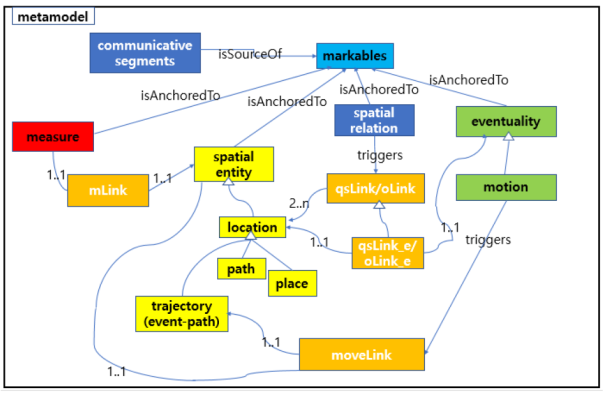

Figure 1: Metamodel of *ISO-Space*

There are three basic *unit element* types in this metamodel: (a) spatial entities; (b) eventualities; and (c) measures. In addition, there are four *relational element* types: (a) QSLINK, qualitative spatial links; (b)

[1]This is an ISO committee draft for the revision of ISO (2014) which restores the original proposal by Pustejovsky et al. (2012) and Pustejovsky and Yocum (2013) that event-paths be treated as a fundamental (complex) entity type triggered by motion events.

OLINK, identifying orientation; (c) MOVELINK, specifying the figure and ground of a movement event;[2] and (d) MLINK, which identifies the metric of a region or distance between regions. These relation types are associated with the SPATIAL_SIGNAL tag.

Qualitative spatial relations (typically within 2D space) can be captured with the relations shown in Table (1) below, from RCC8 (Randell et al., 1992). RCC8 is not, however, able to capture directional or orientational relations and constraints (Freksa, 1992; Frank, 1996; Mossakowski and Moratz, 2012; Zimmermann and Freksa, 1996).

Relation	**Description**
DC	Disconnected
EC	External Connection
PO	Partial Overlap
EQ	Equal
TPP	Tangential Proper Part
TPP_i	Inverse of TTP
NTTP	Non-Tangential Proper Part
NTTP_i	Inverse of NTTP

Table 1: RCC8 Relations.

The goal of this paper is to outline an initial semantics for ISO (2019), a revised version of ISO (2014) focusing on the underlying type structure for the metamodel elements, and how the mapping from annotation structure to interpretation is accomplished.

2 Basic Types and Compositions

The semantics of ISO-Space2019 is formulated on the basis of its abstract syntax, but its interpretation rules apply to the semantic forms which are derived from annotation structures as represented by a concrete syntax. Hence, there are two levels of interpretation that need to be identified when defining a formal semantics of an annotation structure, as applied to linguistic expressions in natural language: language to abstract model; and concrete model to abstract model. In this section, we focus on the first mapping and articulate the underlying semantics of the entities represented in the metamodel in type-theoretic terms and demonstrate the composition of examples within each element type. In the next section, we illustrate the second mapping, from the annotation structure (implemented as a concrete syntactic expression) into the abstract model.

We assume a model with the following basic types, corresponding generally to the elements in Figure 1 above (Kracht, 2002).

(2) a. e, the type of objects
b. i, the type of time points
c. p, the type of spatial points
d. ϵ, the type of events
e. m, the type of measures
f. t, the type of truth values.

Further, following Kracht (2002), we introduce the *group* operator, $^\bullet$, which applies to a type to form a group type, e.g., the group of points, $p^\bullet$. We assume additional types can be constructed with conventional binary type constructors, $\rightarrow$ and $\times$. From these, we can define the standard set of functional types, e.g., $e{\rightarrow}t$, $\epsilon{\rightarrow}t$, $p{\rightarrow}t$, and so on. Further, we assume a semi-lattice of types, where $\sqsubseteq$ is a quasi-ordering on the set of types, such that, for types a,b,c: $a \sqsubseteq b$ and $b \sqsubseteq c$ implies $a \sqsubseteq c$; and $a \sqsubseteq a$. This introduces the *subtyping* relation between types: if $a \sqsubseteq b$, then a is a subtype of b.

[2]The MOVELINK in ISO (2019) is reformulated as outlined in Lee (2016), Pustejovsky and Lee (2017), and Lee et al. (2018).

2.1 Place and Spatial_Entity

The PLACE tag is used for annotating geolocations, such as *Germany* and *Boston*, as well as geographic entities such as lakes and mountains. Further, administrative entities that are registered as geolocations are also tagged as PLACE, e.g., towns and counties. Hence, in the example in (3), the qualitative spatial relation between the two entities is a relation between PLACEs. Both *Gothenburg* and *Sweden* are marked as PLACEs, which we will type as *regions*. A region, r, will be defined at a set of points, $p{\rightarrow}t$. This differs from Kracht (2002), where regions are defined as a subtype of $p^{\bullet}$, where $^{\bullet}$ is a group operator over basic types, but either analysis could be adopted for our present purposes. Further, a qualitative spatial mereotopological relation within RCC8 will be typed as a relation between regions: i.e., $QSLINK : r{\rightarrow}(r{\rightarrow}t)$.

(3) a. [**Gothenburg**$_{pl1}$] is [**in**$_{s1}$] [**Sweden**$_{pl2}$].
b. $[\![\text{Gothenburg}]\!] = G$, $\langle G{:}p{\rightarrow}t\rangle$
c. $[\![\text{Sweden}]\!] = S$, $\langle S{:}p{\rightarrow}t\rangle$
d. $[\![\text{in}]\!] = \lambda y\lambda x[in(x,y)]$, $\langle \text{in}{:}r{\rightarrow}(r{\rightarrow}t)\rangle$
e. $\text{in}(G,S)$

For many spatial relations in language, however, the entities involved are not inherently typed as locations or PLACEs. For example, humans and everyday objects carry a primary type of e, which we subtype or identify here as SPATIAL_ENTITY. When they participate in spatial relations, we assume there is a type coercion function, $\mathcal{L}$, which operates over an entity (or a collection of entities) and returns the spatial region associated with that entity (or entities), i.e., its location in space. Following Klein (1991) will call this the *eigenplace* for the entity (cf. also Wunderlich (1991) and Wunderlich, 1993). The type for this localization operator, $\mathcal{L}$ is: $e{\rightarrow}(p{\rightarrow}t)$. The example in (4) demonstrates how this operator shifts an entity to the type required by the spatial relation, namely r.

(4) a. [**Robin**$_{sne1}$] is in [**Sweden**$_{pl1}$].
b. $[\![\text{Robin}]\!] = R$, $\langle R{:}e\rangle$
c. $[\![\text{Sweden}]\!] = S$, $\langle S{:}p{\rightarrow}t\rangle$
d. $[\![\mathcal{L}(R)]\!] = \lambda x[loc(x,R)]$, $\langle x{:}p, \mathcal{L}{:}e{\rightarrow}(p{\rightarrow}t)\rangle$
e. $[\![\text{in}]\!] = \lambda y\lambda x[in(x,y)]$, $\langle \text{in}{:}r{\rightarrow}(r{\rightarrow}t)\rangle$
f. $\text{in}(\lambda x[loc(x,R)],S)$

The interpretation of SPATIAL_ENTITY in terms of its eigenplace will hold for how objects participate in PATHs as we will see below.

2.2 Paths

We define a path as a subtype of locations (formally regions) that have the additional constraint of being directional, and are often construed as one-dimensional. The notion of a path being introduced or created by an event has its origin in several previous authors, including Cresswell (1978), Jackendoff (1983), and Nam (1995). More recently and more in line with the present specification, we follow the analysis of Mani and Pustejovsky (2012), which is particularly well-suited to the specification in ISO-Space. Formally, paths have been analyzed as sequences of spaces (Nam, 1995) and sequences of vectors (Zwarts and Winter, 2000). Following Nam, let *int* be the type of the interval $[0,1] \subset R$, and p be the type of a spatial point, as defined above. Then a *path*, π, will be that function $int{\rightarrow}p$, which indexes locations on the path to values from the interval [0,1]. Similarly, if *vec* is the type of vectors, then a *vector-based path*, π_v, can be defined as the function $int{\rightarrow}vec$. That is, it indexes the vectors associated with the path to values from the interval [0,1].

(5) a. [**Prague**$_{pl1}$] is on [**the Moldau River**$_{p1}$].
b. [**Boston**$_{pl1}$] is at the end of [**the Mass. Turnpike**$_{p1}$].

In these examples, the qualitative spatial relation introduced by the predication identifies a place as situated within (or on) a path. Hence, the preposition *on* which governs the path-PP, $[_{PP}$ on $[_{NP}$the Moldau River]], carries a more specific type than a general QSLINK relation, namely: $\pi_v \rightarrow (r \rightarrow t)$. The type derivation for (5a) is illustrated below.

(6) a. [**Prague**$_{pl1}$] is on [**the Moldau River**$_{p1}$].
b. $[\![\text{Prague}]\!] = P$, $\langle P{:}p \rightarrow t \rangle$
c. $[\![\text{the Moldau River}]\!] = M$, $\langle M{:}\pi_v \rangle$
d. $[\![\text{on}]\!] = \lambda y \lambda x[on\_path(x,y)]$, $\langle \text{on\_path}{:}\pi_v \rightarrow (r \rightarrow t) \rangle$
e. on_path(P,M)

As sentence (5b) illustrates, end-points of paths can be explicitly mentioned in text. The ISO-Space annotated examples below demonstrate reference to both end-points and mid-points.

(7) a. ... the [**railroad**$_{p1}$] between [**Boston**$_{pl1}$] and [**New York**$_{pl2}$] ...
PATH (`id=p1`, `beginID=pl1`, `endID=pl2`, `form=NOM`)

b. John took the [**road**$_{p1}$] through [**Boston**$_{pl1}$].
PATH (`id=p1`, `midIDs=pl1`, `form=NOM`)

Formally, the expressions introducing end- and mid-point locations are acting as functions from paths to path positions: $\pi_v \rightarrow int$; e.g., given a path $\langle 3,4,5,2,1,8 \rangle$, $end(\pi_v) = 8$.

(8) a. [**Boston**$_{pl1}$] is at the end of [**the Mass. Turnpike**$_{p1}$].
b. $[\![\text{Boston}]\!] = B$, $\langle B{:}p \rightarrow t \rangle$
c. $[\![\text{the Mass. Turnpike}]\!] = MT$, $\langle MT{:}\pi_v \rangle$
d. $[\![\text{end}]\!] = \lambda x[end\_of(x)]$, $\langle x{:}\pi_v, \text{end\_of}{:}\pi_v \rightarrow int \rangle$
e. $[\![\text{on}]\!] = \lambda y \lambda x[on\_path(x,y)]$, $\langle \text{on\_path}{:}\pi_v \rightarrow (r \rightarrow t) \rangle$
f. on_path$(B,MT) \wedge end_of(MT) = B$

As mentioned above, the eigenplace of a SPATIAL_ENTITY can be situated on a path by coercion: namely, $\mathcal{L}$ coerces *John* to his eigenplace, and then the spatial relation predication situates this region onto the path, π_v.

(9) a. [**John**$_{sne1}$] is on [**the road**$_{p1}$].
b. $[\![\mathcal{L}(J)]\!] = \lambda x[loc(x,J)]$, $\langle x{:}p, \mathcal{L}{:}e \rightarrow (p \rightarrow t) \rangle$

3 Events and Paths Generated from Events

The term *event* as it is used in ISO-Space is borrowed directly from ISO-TimeML (ISO, 2012) , and is used as a cover term for situations that *happen*, *occur*, *hold*, or *take place*. Following Davidson (1967) and Parsons (1990), we can represent the event as an individual predicated of an event class (the verb), where the arguments are then related by semantic role relations. It has further been proposed that there is internal structure to events which structurally differentiates the Aktionsarten of Vendler's classes. This has come to be known as *event structure*.[3] On this theory, the subevent structure of the event is explicitly represented in the lexical semantics and subsequent compositional interpretations, giving rise to three basic event structures, STATE, PROCESS, and TRANSITION. The EVENT tag captures ISO-TimeML events that are related to another ISO-Space element by way of a link tag (e.g., a spatial anchoring such as "*sleeping* in the courtyard"). The MOTION tag, on the other hand, identifies those events involving movement of an object through space. All MOTION tags participate in a MOVELINK relation.

There are two basic strategies that languages typically exploit to convey the movement of an object through space (Talmy, 1985): path verb constructions; and manner verb constructions.

[3]Cf. Pustejovsky (1991) and Moens and Steedman (1988).

(10) a. Path Motion: *John arrived at home.*
b. Manner Motion: *John walked.*

In terms of their event structure, path-verbs are transitions while manner verbs are processes. In addition, path verbs are those predicates that *presuppose* a specific path for the moving object (the figure), along with a possible distinguished point or region on this path (the ground), which the figure is moving toward or away from. Manner verbs can be seen as *creating* a path as the motion event unfolds. This is illustrated formally below.

(11) a. Path-presupposing verb (with temporal anchor):
$\lambda y \lambda x \lambda i \lambda e \exists e_1, e_2, p[@_i \text{arrive}(e) \wedge \text{arrive\_act}(e_1, x, p) \wedge DC(e_1, x, y) \wedge \text{arrive\_result}(e_2, x, p) \wedge EC(e_2, x, y) \wedge \text{end}(y, p) \wedge e = e_1 \circ e_2 \wedge e_1 \leq e_2 \wedge e_1 \leq e \wedge e_2 \leq e]$

b. Path-introducing verb (with temporal anchor): :
$\lambda x \lambda p \lambda i \lambda e[@_i \text{walk}(e) \wedge \text{walk\_act}(e, x, p)]$

Path predicates make the change of location explicit in the subevent representation (cf. Pustejovsky (1995). This states that the figure, x, moves along a path, p, represented by the event e. This entails a transition from not being at the ground, e_1, to finally being at the ground, e_2. It further gives the necessary temporal constraints along with the constraint that the ground must be the termination of the path.

The type of the path variable, p, introduced above is no different than that used in the examples in (4)-(9), namely π_v or *int→vec*. The difference, however, is that there is no lexical offset (markable) in the sentences in (10), which can be associated with this path.

Because we are interested in semantically interpreting the annotation structure associated with a linguistic utterance, we will need to distinguish between the concept of *path* encountered above, which is a component part of the domain of space (or a vector space), and this new motion-dependent concept of path: namely, an *event path* is that region of space occupied by a mover throughout an event. For this reason, Lee et al. (2018), following Pustejovsky et al. (2012), suggest that ISO-Space introduce a distinct tag, called an EVENT_PATH. We can type an event path as that path which is associated with an object over time. Assuming the moving object, x, can be represented spatially as its eigenplace, $\mathcal{L}(x)$, the trace of the path created by x is typed as follows: *event path*, π_ϵ, as the function $\epsilon \rightarrow \pi_v$. This is a function from events to the paths they create.

4 Semantic Interpretation of Annotation Structures

In this section we will demonstrate how the concrete syntax of ISO-Space, as deployed over a natural language example, receives an intermediate semantic interpretation, which can then be subsequently interpreted in a model. That is, the semantics of ISO-Space validates each of the annotation structures by mapping it into a semantic form and then interpreting it model-theoretically.

In an XML-based concrete syntax, the two elements `<eventPath>` and `<moveLink>` are implemented each with a list of attribute-value specifications. Each instance of a motion-event triggers an event-path and each event-path is uniquely associated with a motion-event. Such a motion-event is represented by the attribute `@trigger` with a specific value referring to that motion-event associated with an event-path. As a finite path, every event-path has two ends: one is identified as its start and the other, as its end because it is directed. Hence, the attributes `@start, @mids`, and `@end` are required attributes. Their values are *unspecified* if these locations are not explicitly mentioned.[4]

The semantics proposed here maps each of the entity structures into a semantic form and then combines all of the semantic forms compositionally into a final semantic representation based on the associated link structures. Each of the annotation structures is interpreted as a Discourse Representation Structure (DRS), as defined in Kamp and Reyle (1993), through the interpretation function σ, a mapping

[4]Spatial relators such as *from*, *to*, and *through* just define the start, end, and mids of an event-path, without carrying any semantic content. Once the delimiting bounds of an event-path are marked up, the function of spatial relators is discharged.

from the set of entity structures to first-order well-formed expressions, with unbound variables being interpreted as existential or set-denoting. An example annotation structure representation is shown in (12) along with the following σ interpretations.

(12) a. **Dataset**: John arrived in Gothenburg.
Word-segmented: John_{w1} arrived_{w2} in_{w3} Gothenburg_{w4}.

b. **Core annotated**: John_{x1} arrived_{m1} in_{sr1} Gothenburg_{pl1} $\emptyset_{ep1}$.

c. **Annotation Structures**

```
<annotation xml:id="is1" lang="en" aScheme="ISO-Space"/>
{*Entity Structures*:}
<entity xml:id="x1" target="#w1" type="person"/>
<motion xml:id="m1" target="#w2" type="path" tense="past"/>
<sRelation xml:id="sr1" target="#w3" type="endPt-defining"/>
<place xml:id="pl1" target="#w4" form="nam" ctv="city"
   type="ppl" country="sw"/>
<eventPath xml:id="ep1" target="" end="#pl1"/>
{*Link Structure*:}
<moveLink xml:id="mvL1" relType="traverse" figure="#x1"
   ground="#ep1" trigger="#m1"/>
</annotation>
```

(13) a. **Semantic Representation of the Entity Structures**
$\sigma(x_1) = [named(x, John) \wedge person(x)]$
$\sigma(m_1) = [arrive(m) \wedge past(m)]$
$\sigma(pl_1) = [named(l_1, Gothenburg) \wedge city(l_1) \wedge in(l_1, sw)]$
$\sigma(ep_1) = [route(p) \wedge starts(p, <l_0, i_0>) \wedge ends(p, <l_1, i_1>)]$

b. **Semantic Representation of the Link Structure**
$\sigma(mvL_1) = [mover(x,e) \wedge \lambda PP(x)(\sigma(x_1))] \wedge$
$[\lambda PP(p)(\sigma(ep_1)) \wedge traverses(x,p)]$

Interpretations (a) and (b) in (13) show how each of the annotation structures is translated through the interpretation function, σ into a first-order expression. Being a complex structure with IDREFs for the values of its attributes, the link structure has extra λP expressions each of which allows a required variable adjustment.

(14) **Semantic Representation of the Entire Annotation Structure**
$\sigma(is1) = [\sigma(mvL_1) \oplus [\sigma(x_1) \oplus \sigma(m_1) \oplus \sigma(pl_1) \oplus \sigma(t_1) \oplus \sigma(ep_1)]]$
$= [mover(x,e) \wedge named(x, John) \wedge person(x)] \wedge$
$[route(p) \wedge starts(p, <l_0, i_0>) \wedge ends(p, <l_1, i_1>) \wedge$
$[named(l_1, Gothenburg) \wedge city(l_1) \wedge in(l_1, sw)] \wedge$
$[traverses(x,p)]$

The semantic form of (14) is that of the entire annotation structure ($is1$), compositionally obtained from the list of the semantic forms of the entity and link structures, which are given in (13).

5 Conclusion

In this paper, we have outlined an initial semantics for the specification language ISO-Space. We have proposed a type-theoretic interpretation corresponding to the objects and relations in the abstract syntax metamodel. This is then mapped to the interpretation functions which associate the concrete syntactic elements to the semantic interpretations in the model.

John arrived in Gothenburg.				
Syntax			**Semantics**	
ID	**TARGET**	**ANNOTATION**	**semTYPE**	**semFORM**
x1	John	`type="person"`	$x : e$ (entity)	$named(x, John)$ $person(x)$
m1	arrived	`tense="past"`	$e : event$ (event)	$arrive(e)$ $past(e)$
s1	in			
pl1	Gothenburg	`ctv="city"`	$l : r$ (region)	$named(l_2, Gothenburg)$ $city(l_2)$
ep1	∅	`start="unknown"` `end="pl1"`	$p : \pi_v$ (path)	$starts(p, < l_1, i_1 >)$ $ends(p, < l_2, i_2 >)$
	.	.	t (truth-value)	ϕ

mvL1		`figure="x1"`
		`ground="ep1"`
		`relType="traverses"`
				$mover(x, e)$
				$route(p)$
				$traverses(x, p)$

Table 2: Semantics based on Abstract Syntax

References

Cresswell, M. (1978). Prepositions and points of view. *Linguistics and Philosophy 2*, 1–41.

Davidson, D. (1967). The logical form of action sentences. In N. Rescher (Ed.), *The Logic of Action and Decision*, pp. 81–95. University of Pittsburg Press.

Frank, A. U. (1996). Qualitative spatial reasoning: Cardinal directions as an example. *International Journal of Geographical Information Science 10*(3), 269–290.

Freksa, C. (1992). Using orientation representation for qualitative spatial reasoning. In A. Frank, I. Campari, and U. Formentini (Eds.), *Theories and Methods of Spatio- temporal Reasoning in Geographic Space: Proceedings of the International Conference GIS - From Space to Territory*, Pisa, Italy, pp. 162–178.

ISO (2012). *ISO 24617-1:2012(E) Language resource management - Semantic annotation framework - Part 1: Time and events (ISO-TimeML)*. International Organization for Standarization, Geneva. Note: ISO Project leader: Kiyong Lee; Editors: James Pustejovsky (chair), Branimir Boguraev, Harry Bunt, Nancy Ide, and Kiyong Lee, ISO/TC 37/SC 4/WG 2.

ISO (2014). *ISO 24617-7:2014(E) Language resource management - Semantic annotation framework - Part 7: Spatial information (ISO-Space)*. International Organization for Standardization, Geneva. Note: ISO Project leaders: James Pustejovsky and Kiyong Lee, ISO/TC 37/SC 4/WG 2.

ISO (2019). *ISO/CD 24617-7:2014(E) Language resource management - Semantic annotation framework - Part 7: Spatial information (ISO-Space)*. International Organization for Standardization, Geneva. Note: ISO Project leaders: James Pustejovsky and Kiyong Lee, ISO/TC 37/SC 4/WG 2.

Jackendoff, R. (1983). *Semantics and Cognition*. MIT Press.

Kamp, H. and U. Reyle (1993). *From Discourse to Logic: Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Kluwer Academic Publishers, Dordrecht.

Kracht, M. (2002). On the semantics of locatives. *Linguistics and philosophy 25*(2), 157–232.

Lee, K. (2016). An abstract syntax for ISO-Spac with its `<movelink>` reformulated. In H. Bunt (Ed.), *Proceedings of the Twelfth Joint ACL - ISO Workshop on Interoperable Semantic Annotation (ISA-12)*, pp. 28–37. Workshop at LREC2016, 28 May 2016, Portorož, Slovenia.

Lee, K., J. Pustejovsky, and H. Bunt (2018). Revising ISO-Space and the role of the movement link. In H. Bunt (Ed.), *Proceedings of the Fourteenth Joint ACL - ISO Workshop on Interoperable Semantic Annotation (ISA-14)*, pp. 35–44. Workshop at COLING2018, August 25, 2018, Santa Fe, New Mexico, USA.

Mani, I. and J. Pustejovsky (2012). *Interpreting Motion: Grounded Representations for Spatial Language*. Oxford University Press.

Moens, M. and M. Steedman (1988). Temporal ontology and temporal reference. *Computational Linguistics 14*(2), 15–28.

Mossakowski, T. and R. Moratz (2012). Qualitative reasoning about relative direction of oriented points. *Artificial Intelligence 180*, 34–45.

Nam, S. (1995). *The Semantics of Locative Prepositional Phrases in English*. Ph. D. thesis, University of California at Los Angeles.

Parsons, T. (1990). *Events in the Semantics of English. A Study in Subatomic Semantics*. Cambridge, MA: MIT Press.

Pustejovsky, J. (1991). The generative lexicon. *Computational Linguistics 17*(4).

Pustejovsky, J. and K. Lee (2017). Enriching the notion of path in ISO-Space. In H. Bunt (Ed.), *Proceedings of the Thirteenth Joint ACL - ISO Workshop on Interoperable Semantic Annotation (ISA-13)*, pp. 134–139. IWCS 2017 Workshop, Montpellier, France, September 19, 2017.

Pustejovsky, J., J. Moszkowics, and M. Verhagen (2012). The current status of ISO-Space. In H. Bunt (Ed.), *Proceedings of the Seventh Joint ACL–ISO Workshop on Interoperable Semantic Annotation (ISA-7), SRSL3 Workshop on Semantic Representation for Spoken Language, and I2MRT Workshop on Multimodal Resources and Tools*, pp. 23–30. LREC 2012 Workshop, Istanbul, Turkey.

Pustejovsky, J. and Z. Yocum (2013). Capturing motion in ISO-SpaceBank. In H. Bunt (Ed.), *Proceedings of the Ninth Joint ACL–ISO Workshop on Interoperable Semantic Annotation (ISA-9)*, pp. 25–34. IWCS 2013 Workshop, Potsdam, Germany.

Randell, D. A., Z. Cui, and A. G. Cohn (1992). A spatial logic based on regions and connection. *KR 92*, 165–176.

Talmy, L. (1985). Lexicalization patterns: semantic structure in lexical forms. In T. Shopen (Ed.), *Language typology and semantic description Volume 3: Grammatical categories and the lexicon*, pp. 36–149. Cambridge University Press.

Wunderlich, D. (1991). How do prepositional phrases fit into compositional syntax and semantics? *Linguistics 29*(4), 591–622.

Zimmermann, K. and C. Freksa (1996). Qualitative spatial reasoning using orientation, distance, and path knowledge. *Applied Intelligence 6*(1), 49–58.

Zwarts, J. and M. Winter (2000). Vector space semantics: A model-theoretic analysis of locative prepositions. *Journal of Logic, Language, and Information 9*, 169–211.

Evaluating distributional representations of verb semantic selection

Elisabetta Jezek
Università degli Studi di Pavia
jezek@unipv.it

Edoardo Maria Ponti
University of Cambridge
ep490@cam.ac.uk

Bernardo Magnini
Fondazione Bruno Kessler
magnini@fbk.eu

Abstract

The purpose of this paper is to evaluate whether distributional techniques applied to lexical sets, i.e. the set of fillers of verb argument slots, constitute a useful heuristic to model verb semantic selection. To achieve this purpose, we extract the word vectors corresponding to our lexical set vocabulary from the word2vec distributional semantic model, and then perform *k*-means clustering on these. We focus on verbs undergoing the causative/inchoative alternation as a case study, as they offer an interesting challenge due to the theoretical assumption that the lexical sets of the transitive Object (O) and the intransitive Subject (S) overlap. We analyze the obtained clusters from a qualitative point of view, calculate the prototype vector based on the cluster centroid, and evaluate them against the human judgments on verb semantic selection acquired from a lexical resource. We present an in-depth linguistic analysis of the Italian verb *suonare* 'to ring, to play'. The analysis demonstrates that automatically obtained clusters and human judgments based on manual clustering match closely, although the centroids appear not to be systematically the best indicators of the cluster semantics, and metonymic uses leads to incorrect automatic analysis.

1 Introduction

Empirical attempts to define semantic constraints imposed by verbs on their arguments have relied, among others, on lexical sets, that is, the group of fillers that occupy the argument positions of a verb in a corpus, such as {*book*, *post*, *story*, *review*, *blog*, *report*, *text* ...} for the object slot of *to read*. [1] Semantic constraints on argument selection may receive an empirical validation from distributional semantics techniques applied to lexical sets. In this paper, we propose a methodology to ground semantic selection based on quantization of low-dimensional word vectors of argument fillers, and evaluate it against human judgements acquired from a lexical resource.

As a test bed for the methodology, we focus on verbs exhibiting the so-called causative-inchoative alternation, which offer an interesting challenge due to the theoretical assumption that the lexical sets of the Object of the causative variant (O) and the Subject of the inchoative one (S) (cf. section 4) overlap. We examine the lexical sets of both S and O. We extract word vectors, perform *k*-means clustering, analyze the results qualitatively and evaluate them against human judgements acquired from a lexical resource. We show that the fillers of S and O appear to be both intersective and non intersective (cf. section 3); in both cases, they can be grouped together into several homogeneous and well-defined semantic categories. Our research represents a contribution to the evaluation of the effectiveness of vector quantization with *k*-means in the identification of (sub)categories with homogeneous conceptual content, starting from lexical sets of argument positions.

The structure of the paper is the following. Section 2 briefly points to related work, section 3 outlines the framework, section 4 introduces the case study, section 5 presents the data and the method, and section 6 illustrates the experiment and the results obtained. The conclusions are presented in section 7.

[1]The notion of lexical set we rely on in this study was firstly introduced in computational lexicography by Hanks (1996). In his work, Hanks conceives lexical sets in relation to verb senses. For our current purposes we will not consider verb sense distinctions in this paper and will postpone the analysis of polysemy effects to subsequent work.

2 Related work

Our linguistic proposal relates to attempts in the computational linguistics community to assess selectional preferences and estimate thematic role fit for verbs within the paradigm of distributional semantics, in particular to the work of Erk (2007), Baroni and Lenci (2010), and more recently Greenberg et al. (2015), Sayeed et al. (2016), Chersoni et al. (2017), among others. Particularly, in Greenberg et al. (2015) an interesting methodology for clustering filler vectors is proposed, that takes into account verb sense distinctions. Most of these works focus on the notion of semantic plausibility, and address the problem of assessing verb semantic preferences on their arguments in terms of cognitive expectation and psycholinguistic processing. In a complementary way, in our work we approach verb semantic selection in terms of semantic categories, and focus on heuristics to identify such categories instead of on assigning thematic fit scores to single lexical items filling the argument positions.

3 Clustering Low-Dimensional Vectors of Argument Fillers

As referenced above, the strategy we rely on for grounding verb semantic selection on empiric evidence comes from distributional semantics. Mikolov et al. (2013) devised word2vec, a well-known efficient technique representing words from a huge corpus as vectors of *n* real numbers (a sequence of coordinates in a *n*-dimensional space). Its algorithm CBOW (continuous bag of words) yields a vector model trained through unsupervised machine learning, and based on the assumption that the meaning of a word can be inferred by its neighbour words. This model has the relevant property of representing as geometric distance the similarity in meaning.

The vector model created by these means can be fed into a clustering algorithm: this operation is called vector quantization. In our work, clusters serve the purpose of validating and providing insight about verb semantic selection. We inspect the lexical membership of the clusters obtained applying *k*-Means (MacQueen, 1967) on the vectors of the lexical sets harvested from a parsed corpus (see section 5). Our goal is to verify whether this membership can be taken as representative of the semantic constraints that verbs place on their arguments, in which case the methodology can be used to empirically model verb semantic selection.

4 Case Study: Causative/inchoative Verbs

The idea of taking advantage of word vectors and clustering procedures to model verb semantic selection needs to be tested by applying it on a specific domain. As a case of study, we focus on verbs showing a causative/inchoative alternation in Italian. In all languages, many pairs of verbs can be found which lexicalize a pair of events, of which one is the cause of the other, as in *to kill* (cause) and *to die* (result). Lemmas are not always distinct, instantiating a suppletive pattern; in several cases, a single form in the lexicon can be polysemous between a sense that encodes the cause, and a sense that lexicalizes the result. This is the well-known case of causative/inchoative alternating verbs such as *to break*, as in “Mary broke the key" (cause) and “The key broke” (result) (Levin, 1993; Jezek, 2003).[2] We focus on this verb class because one can expect that verb alternations preserve semantic selection, thus alternations provide an opportunity to test to what extent semantic selection can be characterized by distributions semantics, S/O alternation in particular.

For the purposes of this study, we consider a list of causative-inchoative alternating verbs that display a wide cross-linguistic variation. The list is inspired by Haspelmath et al. (2014), where the verbs are ranked from most causative-prominent to most inchoative-prominent based on frequency and morphosyntactic coding of the two variants. The list includes the following verbs: *affondare* 'to sink', *alzare* 'to raise/rise', *aprire* 'to open', *aumentare* 'to improve', *bollire* 'to boil', *bruciare* 'to burn', *chiudere* 'to

[2] More precisely, the causative variant describes both the cause and the result; see the semantic incongruity of “*Mary broke the key and the key did not break”.

close', *congelare* 'to freeze, *connettere* 'to connect', *dividere* 'to split', *finire* 'to finish', *girare* 'to turn', *raccogliere* 'to gather', *riempire* 'to fill', *rompere* 'to break', *sciogliere* 'to melt', *scuotere* 'to rock', *seccare* 'to dry', *suonare* 'to ring/to play', *svegliare* 'to wake up', *uscire* 'to go out'. In a previously published study (Ponti et al. (2017), we have analyzed these verbs and shown that a higher cosine distance of the centroid vectors of their lexical sets for S and O matches inchoative-prominence in Haspelmath's scale.

5 Data and Method

In this section, we present the data and the method for extraction, vectorization and clustering of fillers belonging to lexical sets of the target verbs. The first step consists in obtaining the fillers for the argument position of each verb. Our data are sourced from one of the widest freely available corpus for Italian, ItWac: it is obtained by crawling texts from the Italian domain in the web using medium frequency vocabulary as seeds and contains 1,585,620,279 tokens (Baroni et al., 2009). From this resource, a sample of 2,116,648 sentences was extracted. This sample was further filtered based on sentence length (sentences longer than 99 tokens were discarded for computability's sake) and enriched with syntactic information through the MATE-tools parser (Bohnet, 2010). LAS scores for the relevant dependency relations were: 0.751 with dobj (direct object), 0.719 with nsubj (subject), 0.691 with nsubjpass (subject of a passive verb). The parser model was trained on the Italian treebank in the HamleDT repository (Zeman et al., 2014; Bosco et al., 2013), whose annotation style is based on Stanford Dependencies. At the end, sentences in the sample amounted to 2,029,454 items. In the parse trees, the target predicate verbs were identified together with their main lexically full arguments. The lemmas of these verbs and the forms of the respective arguments were stored in a database. Argument fillers were grouped based not on their grammatical relation, but rather on their semantic macro-roles according to Dixon (1994): subjects of transitive verbs (A), subjects of intransitive verbs (S) and objects (O). Crucially, the subjects of verb forms accompanied by the *si*-clitic were treated as S. Subjects of verbs inflected in the passive voice were treated as O instead. These operations resulted in a database structured as a list: in each row a verb is followed by the fillers recorded in a particular text occurrence classified by macro-roles. The database was later collapsed by verb lemma so that each became associated to three sets of fillers (one per macro-role). Each of these sets represents a corpus-derived lexical set.

Afterwards, each of the fillers was mapped to a vector. This operation relied on a pre-trained word embedding made available by Dinu et al. (2015). The authors created it employing word2vec on the itWaC corpus with the following settings: CBOW with negative sampling algorithm, 300 dimensions, window of 10 tokens, with sub-sampling. This resulted into a vector model inclusive of fillers only. The results of the experiment in the rest of the paper refers to this sub-model.

The comparison of (sets) of vectors was achieved through their cluster membership. Vector quantization is the operation of defining *k* clusters in a model and then assessing vector membership. This process abides to the following rules: every vector is assigned to one and only one cluster. Moreover, the globally optimal solution is such that a measure of variance is minimised. This measure is usually the within-cluster sum of squares (WCSS), i.e. the sum of Euclidean distances of the vectors of a cluster from its centre (i.e. its mean). The algorithm employed to perform vector quantization was *k*-Means (MacQueen, 1967). The algorithm converges into a local optimum by initialising *k* arbitrary means inside the model. Then it carries on iteratively two steps: firstly it assigns each vector to these means minimising the WCSS. Secondly, it calculates the new means. The algorithm stops when no mean shift becomes possible any more.

Estimating the best *k* can be performed heuristically with the so-called elbow method. Trying several possible values for *k*, the one is chosen that guarantees the best trade-off in minimising the cluster number and maximising their internal similarity. The application of the elbow method to the fillers-only model yielded the result in Figure 1: the elbow is around $k = 377$.

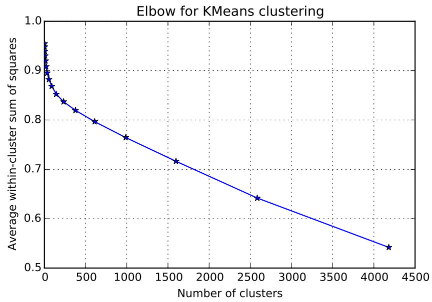

Figure 1: Elbow of Ks (number of clusters) against WCSS (internal similarity).

6 Experiment and Results

In order to verify whether the obtained clusters are semantically homogeneous and whether the lexical sets of S and O for causative/inchoative verbs overlap, we first indexed every filler of a given lexical set of a target verb to its cluster, and then created the Euler-Venn diagrams of the two sets of clusters to which the fillers of S and O point. Only clusters with at least five members and fillers at least three characters long were included, in order to filter out noise.

In the following we centre the discussion on the Italian verb *suonare* ('to play' / 'to ring'), which is particularly interesting as regards the intersections between S and O (see section 4). While an in-depth discussion of a single verb may appear questionable from a methodological point of view, it allows us to reach the detail of linguistic analysis that we deem necessary for the evaluation of the methodology, which is the main goal of this contribution. We report the results for this verb and provide a qualitative analysis that confirms the absence of perfect overlap between the lexical sets for S and O (Montemagni et al., 1995). We detected 9 cluster that are relevant for the verb *suonare* (see Figure 2): cluster 24, 58, 143, 157, 193, 257, 278, 285 and 325. In Figure 2, the three blue bar plots represent the cardinality of these clusters (i.e. the number of fillers of the verb in the specific argument position that are members of the cluster) in both the intersection and the two differences of the Euler-Venn diagram (S-O stands for Subject-only fillers and O-S for Object-only fillers). It is evident how the intersection is dominated by clusters 58 and 143.

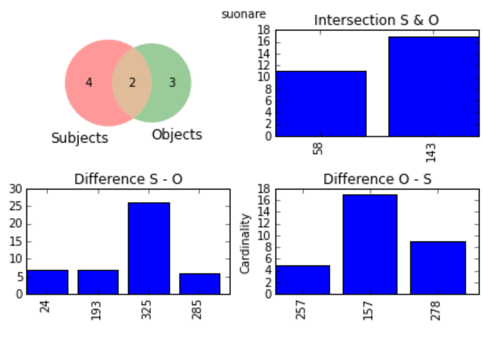

Figure 2: 9 relevant clusters for *suonare*. Intersection of their fillers (S and O), Subject only (difference S-O), and Object only (difference O-S).

Each cluster is identified by a number and has no given label for its semantics. In order to characterize clusters semantically, we first identified the filler with the vector closest to the cluster centroid ("prototype" in Table 1). Closeness is defined based on cosine distance with respect to the centroid.[3] Then, we manually assigned a tentative "semantic label": for example the label *sounds* for the prototype *brontolio* 'grumble' of cluster 157. This step was done by one author and later discussed and agreed with with the others. Thus clusters 58 and 143 in the intersection are characterised by discs (artefacts on which music

[3] See details in Ponti et al. (2017).

cluster	prototype	closeness	label
24	storie	0.708	stories
58	album	0.789	discs
143	clarinetto	0.736	instruments
157	brontolio	0.807	sounds
193	rockabilly	0.758	performers
257	palermo	0.686	towns
278	cornamuse	0.726	instruments
285	aggettivo	0.736	language
325	scemenza	0.801	judgements

Table 1: Cluster centroids (prototypes), closeness (cosine distance) and manually assigned semantic labels.

is recorded) and musical instruments. For reasons of space, we cannot provide here the full list of fillers for all 9 relevant cluster of *suonare*, but they can be found in the Appendix A at the end of the paper.

Other clusters collect fillers that are not intersective. In S - O (Subject only) this is the case for *parole* 'words' (24), *nome* 'name/noun' and *termine* 'term' (285), *band* 'band' and *chitarrista* 'guitarist' (193). Clusters 24 and 285 are explained by the fact that *suonare* has a meaning corresponding to 'to seem', 'to appear' (as in "questa storia (mi) suona strana" 'this story sounds strange (to me)') that is available for the intransitive variant only. 193 is due to the fact that *suonare* undergoes an alternation such that the transitive object can be omitted, and the subject remains agentive even with a mono-argumental use of the verb ("mentre il chitarrista suonava..." 'while the guitarist was playing...'); the lack of object leads to a misclassification of A's as O's.[4] The inspection of the members of cluster 325 suggests that also this cluster is related to the figurative use of the verb observed for clusters 24 and 285.

As for O - S (Object only), the words that can behave only as objects are *brontolio* 'grumble' (prototype of 157, denoting a type of sound), but also *campanello* 'doorbell', *clacson* 'car horn', *sirene* 'sirens', which denote sound-producing artifacts: both these word types are grouped in cluster 157. Cluster 278 has *cornamuse* 'bagpipes' as prototype, which suggest a semantic label "instruments"; however, an inspection of its members reveals that the majority of the words denote in fact musical compositions, such as *cantilena* 'singsong', *litanie* 'litanies', *ritornello* 'refrain' and *canzonetta* 'jingle'; it appears that in this case the centroid is not the best indicator of the semantics of the cluster. Finally, cluster 257 is an irrelevant cluster probably due to a systematic parsing error (object instead of locative complement).

The qualitative analysis of *suonare* demonstrates that vector quantization of argument fillers delivers clusters that are semantically motivated and can be modeled, albeit approximately, as semantic categories. The centroid is not always a good semantic indicator of such categories. The analysis also reveals that exceptions to the perfect overlap between S and O (i.e. the fillers in the differences) are present,[5] and that they are not isolated, but rather are organized semantically themselves.

7 Evaluation

We compare the qualitative results of our experiment for the verb *suonare* with the human judgements on the semantic selection of the verb collected within the T-PAS resource (Jezek et al., 2014). These human judgements are corpus-based, that is, they are generalizations that the annotators express while manually inspecting and clustering occurrences of the verb in the corpus. They are expressed in terms of semantic types. Annotators choose among a list of 180 types. In the resource, for the Object slot of *suonare* there are three judgments for three distinct verb senses: `Musical Instrument` for the sense 'produce music from', `Sound Maker` for the sense 'cause to produce a sound', and `Musical`

[4] This is part of the problem that verbs are not disambiguated by sense in the approach we present here, see note 1.

[5] The graphs in Figure 2 show a very scant intersection between S and O for *suonare*; however, this is mainly due to the small sample. The more it is enlarged, the more the intersection expands.

`Composition` (e.g. *canzone* 'song') for the sense 'play'. For the Subject position, there is `Human` for the sense 'performs music, performs in musical performances, draws sound from a musical instrument'; `Anything` (the top type of the system) for the sense 'seem, appear', and again `Sound Maker` and `Musical Instrument` for the sense 'emit sound'.

Overall, human judgments based on manual clustering and automatically obtained clusters match closely. There is matching between the judgment `Musical Instrument` in both Object and Subject position with cluster 143, which corresponds to the dominant intersective cluster in Figure 2. The judgment `Sound Maker` is not directly matched by any cluster, although as observed in section 6 the inspection of the members of cluster 157 revealed several word of such kind (*campanello* 'doorbell'). Cluster 157 has *sound* as prototype, which is not matched by any judgment. In this case, the centroid appears not to be a good predictor of the cluster semantics. Several words corresponding to the judgment `Musical composition` are found in cluster 58 (*canzone* 'song', *brano* 'piece of music'), whose prototype is *album*. We think that in this case the mismatch between judgment and prototype can be explained by assuming that album and similar objects are used metonymycally for the musical compositions recorded on them. For the Subject only, cluster 193 is matched by the judgement `Human` for the sense 'play music' (note that the cluster label *performers* is more fine-grained), while the remaining clusters (24, 325, 285) are all matched by the semantic type `Anything` for the sense 'to seem, to appear' (see section 6). The evaluation thus reveals that clustering is a viable methodology to model semantic selection provided that we are aware that the centroid is not systematically the best predictor of the cluster semantics and that metonymic uses may lead to incorrect semantic analysis.

8 Conclusions

The purpose of this work was to evaluate whether distributional techniques applied to lexical sets of verb argument positions constitute a useful heuristic to model verb semantic selection. We focused on causative-inchoative verbs that offered the additional challenge of validating the theoretical assumption that the lexical sets of S and O overlap. We proposed an in-depth analysis of the It. verb *suonare* 'to ring, to play', and performed the evaluation exploiting an existing lexical semantic resource that contains human judgments on verb semantic selection based on manual clustering of occurrences from the corpus.

The results we obtained support the view that modeling the words that make up the lexical sets as vectors and quantizing them is a viable methodology to characterize verb selectional constraints in terms of homogeneous semantic categories. Moreover, they provided evidence that there are exceptions to the perfect overlap between the lexical sets of S and O with causative/inchoative verbs, and that these exceptions are not isolated, but rather are organized semantically themselves. A critical aspect that our experiment contributes to highlight is that the centroids appear not to be systematically good predictors of the cluster semantics, casting doubts on their use for this purpose in distributional semantic models. Finally, our analysis brings attention to metonymic uses that introduce noise in the automatic clustering procedure.

Further research should resort to an enhanced database with a wider sample, a more accurate parsing (and hence filler identification), and sense disambiguation for polysemous word forms (Grave et al. (2013), Greenberg et al. (2015)), possibly exploiting multi-sense embeddings.

Acknowledgements

This work has been supported by the Italian PRIN project 20159M7X5P_002 *Transitivity and Argument Structure in Flux*. We would like to thank the reviewers for their helpful comments. All remaining errors and infelicities are our own.

References

Baroni, M., S. Bernardini, A. Ferraresi, and E. Zanchetta (2009). The wacky wide web: a collection of very large linguistically processed web-crawled corpora. *Language resources and evaluation 43*(3), 209–226.

Baroni, M. and A. Lenci (2010). Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics 36*(4), 673–721.

Bohnet, B. (2010). Very high accuracy and fast dependency parsing is not a contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pp. 89–97. Association for Computational Linguistics.

Bosco, C., S. Montemagni, and M. Simi (2013). Converting italian treebanks: Towards an italian stanford dependency treebank. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pp. 61–69. Citeseer.

Chersoni, E., E. Santus, P. Blache, and A. Lenci (2017). Is structure necessary for modeling argument expectations in distributional semantics? *arXiv preprint arXiv:1710.00998*.

Dinu, G., A. Lazaridou, and M. Baroni (2015). Improving zero-shot learning by mitigating the hubness problem. *Workshop Contribution at ICLR 2015*.

Dixon, R. M. (1994). *Ergativity*. Cambdrige: Cambridge University Press.

Erk, K. (2007). A simple, similarity-based model for selectional preferences. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pp. 216–223.

Grave, E., G. Obozinski, and F. Bach (2013). Hidden markov tree models for semantic class induction. In *CoNLL-Seventeenth Conference on Computational Natural Language Learning*.

Greenberg, C., A. Sayeed, and V. Demberg (2015). Improving unsupervised vector-space thematic fit evaluation via role-filler prototype clustering. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 21–31.

Hanks, P. (1996). Contextual dependency and lexical sets. *International Journal of Corpus Linguistics 1*(1), 75–98.

Haspelmath, M., A. Calude, M. Spagnol, H. Narrog, and E. Bamyaci (2014). Coding causal–noncausal verb alternations: A form–frequency correspondence explanation. *Journal of Linguistics 50*(03), 587–625.

Jezek, E. (2003). *Classi di verbi tra semantica e sintassi*. Pisa: Edizioni ETS.

Jezek, E., B. Magnini, A. Feltracco, A. Bianchini, and O. Popescu (2014). T-pas: A resource of corpus-derived types predicate-argument structures for linguistic analysis and semantic processing. In *Proceedings of LREC 2014 (Reykjavik, May 26-31)*.

Levin, B. (1993). *English Verb Classes and Alternations*. Chicago: The University of Chicago Press.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, Volume 1, pp. 281–297.

Mikolov, T., I. Sutskever, K. Chen, G. S. Corrado, and J. Dean (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pp. 3111–3119.

Montemagni, S., N. Ruimy, and V. Pirrelli (1995). Ringing things which nobody can ring. a corpus-based study of the causative-inchoative alternation in italian. *Textus online only. 8 (1995), N. 2, 1995 8*(2), 1000–1020.

Ponti, E., E. Ježek, and B. Magnini (2017). Distributed representations of lexical sets and prototypes in causal alternation verbs. *Italian Journal of Computational Linguistics 3*(1), 25–42.

Sayeed, A., C. Greenberg, and V. Demberg (2016). Thematic fit evaluation: an aspect of selectional preferences. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pp. 99–105.

Zeman, D., O. Dušek, D. Mareček, M. Popel, L. Ramasamy, J. Štěpánek, Z. Žabokrtskỳ, and J. Hajič (2014). Hamledt: Harmonized multi-language dependency treebank. *Language Resources and Evaluation 48*(4), 601–637.

A Numbers and Fillers of the Clusters

24 corrispondenze lettere paginette guide cronologie battute collane mappe ricette schede commedie favole avventure parodie descrizioni citazioni discografie frasi imitazioni sottotrame cartine filastrocche traduzioni pagelle caricature cartoline letture opere copertine aneddoti barzellette bufale illustrazioni leggende parabole vignette biografie scritture pagine storie recensioni chicche leggendole parole chiose cronache memorie righe autobiografie note storielle didascalie postille voci icone tavole comparse rubriche narrazioni puntate versioni interviste fiabe

58 ristampe dischi singles discografiche canzone classifiche label cantautore reunion promo reprise cofanetto records singolo bootleg cover cd live vasco scaletta prozac autoproduzione lp esordio greatest loser pearl canzoni debutto album missaggio demo disco compilation fans vinile remix etichetta videoclip soundtrack major itunes discografici track pausini brano brani titletrack mescalina fan

143 gregoriano arrangiamenti violoncello flauto vocale vocali recitativi violinista duetto duetti intonazione arrangiamento compositore musicale violoncellista batteria fonico quintetto improvvisazione violino suoni suono suona sonoro archi organista ascoltatore quartetto arie polifonia percussioni percussione ottoni contrappunto soprano clarinetto djembà djembe melodia musicista ottavino arrangiatore baritono acustica solfeggio partiture partitura congas diapason orchestra pianista ottave chitarre ensemble tromba ghironda sabar spartito oboe musiche tastiere corni conservatorio sintetizzatore armoniche chitarra pianoforte organetto ugola accordatore musica mandolino orchestrali griot

157 chiasso muto grida tuoni tuono brontolio folla cicale agitarsi fischi strepiti baccano clacson fracasso rintocchi fruscio grido frastuono eco silenzio brusio sbadiglio campanello ululati ululato sordo urlando campanacci orecchio gemito gemiti andirivieni sommesso ronzio timpani ruggito ruggiti squillo fragore fragori gorgoglio mormorio rumore rumori urla urlo rantolo stridore udito miagolio tonfo tumulto spari rincorrersi voce sospiri detonazioni singhiozzo campane fischio risa udiva muezzin sirena sirene lamento lamenti tam battito battiti

193 roll zappa vox rapper smiths blink tastierista wave metallari pogo rockabilly sound cantautori jazz bassista mood gang chitarrista mambo nada rockettaro dj mariachi radiohead rasta hippies trip folk iggy psichedelia grunge reggae metallaro ballad band trance paranoid kid caparezza rock boys pop underground rave freak session orecchiabilità combo sequencer boogie linkin punk marlene crew blues yuppie batterista techno batteristi fusion dance crossover cantante duo fugazi beat daft revival rap metal queen trio vocalist ballads rocker

257 pula novara bologna ostia prato giglio capri azzurro foggia colomba pisa roma cosenza catania aquila messina mosca bema modica livorno lecce ambra ferrara reggio napoli campanella mannelli gigli palermo cuneo nola tor rosa aquilone brindisi cesena proteo campana falcone fiesole piacenza vallo pistoia meridiana

278 nenie cumbia danzare struggenti litanie rumba suonatori karaoke valzer cantilena ritornello cantavo balere macarena strofe strofa motivetti danze pizzica bans ritmi jingle canzoncine canto fado campanelli coro coriste suonerà saltarello ibro ritmo carillon cetre ninna filastrocca tormentoni minuetti clave sonagli canti grammelot cori sottofondo cantare tamburi refrain adagio balla canzonette canzonetta medley inno ballata midi inni cantando cantavamo cantata cantato cornamusa

285 nome soprannome aggettivo maiuscola motto toponimo termine denominazioni denominazione nomignolo pseudonimo acronimo patronimico frase gergo accezione locuzione virgolette n.d.r. vocabolo suffisso parentesi abbreviazione terminologia altisonante eufemismo temine parola espressione dicitura prefisso sigla dizione neologismo epiteto cognome appellativo

325 vagonata scommessa tiritera ragione menata fesseria iattura frottola sparata ovvietà mistificazione esclamazione cosa idiozia gaffe peccatuccio sfortuna forzatura bugia battuta stupidaggine indiscrezione vigliaccata pecca scortesia sfiga torto goduria svista corbelleria buonafede incontrario scusante ideuzza stortura mostruosità roba mazzata diceria parolina stronzata peggior schifezza gazzarra chicca mania immonda boutade divagazione sbornia pagliuzza caricatura piaga scemenza fantapolitica incolpa aberrazione sciocchezza falla sproposito barzelletta nomea favoletta jattura furbata stonatura bufala mossa arcinoto colpa ciliegina spocchiosa patacca parolaccia scempiaggine fandonia sceneggiata lamentela figuraccia stranezza porcheria bestemmia guastafeste regoletta dietrologia controprova faccenda chiacchiera madornale balordaggine provocazione esagerazione fregatura verità postilla cretinata disgrazia seccatura cazzata cantonata beffa criticarla stroncatura scaramanzia pernacchia paccottiglia finta frecciata scorciatoia caciara burla boiata assurdità scappatoia robaccia cagnara buffonata

A Semantico-Syntactic Approach to Event-Mention Detection and Extraction In Hindi

Jaipal Singh Goud
IIIT Hyderabad
jaipal.singh@research.iiit.ac.in

Pranav Goel
IIIT Hyderabad
pranav.goel@research.iiit.ac.in

Alok Debnath
IIIT Hyderabad
alok.debnath@research.iiit.ac.in

Suhan Prabhu
IIIT Hyderabad
suhan.prabhuk@research.iiit.ac.in

Manish Shrivastava
IIIT Hyderabad
m.shrivastava@iiit.ac.in

Abstract

This paper introduces a gold standard event-annotated dataset of 810 Hindi news article as well as a set of comprehensive guidelines for detecting and annotating events in Hindi. We present our linguistically motivated guideline development process, with a focus on annotator friendliness, that can be replicated for event mention detection for most Indo-Aryan languages. The paper highlights the challenges of detecting event mentions in Hindi given the unique semantic constraints on the syntactic apparatus used for denoting events. Our work as a whole also establishes a language agnostic pipeline for the development of an event annotated corpora and event detection guidelines.

1 Introduction

Event detection is an important problem in both natural language processing and information retrieval. It has been extended to multiple domains and is now being tackled for languages other than English. In this paper, we address the problem of event detection in Hindi. This task is relatively unexplored for Hindi, partly because of syntactic features such as relative free word order and nature of verb adjuncts and partly due to the lack of structured data. The syntax of the language increases the number of options available for expressing structured information, hence significantly complicating how chunks of text can be semantically interpreted. Therefore, our approach to event detection is based on defining events in a semantico-syntactic idea, meaning that certain clause, phase and sentence structures are *eventive* in nature.

This task is an important one given the constant rise in volume of Hindi data being produced and consumed. A recent report shows that Hindi content consumption on the web is growing at a rate of 94% per year as compared to English's 19% [1]. From the perspective of extracting latent information, defining events as semantico-syntactic objects allows for extracting relations between both events and entities, that are not explicitly mentioned. The extracted event information can then be readily consumed by downstream tasks such as question answering, temporal and causal inference detection and summarization.

[1]https://economictimes.indiatimes.com/tech/internet/hindi-content-consumption-on-internet-growing-at-94-google/articleshow/48528347.cms

Therefore, in this paper, we first define events in Hindi and explore the challenges involved in detection of these event in Hindi news text. We then define a schema and a set of guidelines for event detection in Hindi. Here, the schema refers to the various components of the annotated textual span, such as the event trigger, nugget and core, while the guidelines define the mechanism for the annotation of events based on syntactic characteristics of the schema. The guidelines, while inspired by TimeML (Pustejovsky et al., 2003) in format and manner, are **not** a direct adaptation of the TimeML guidelines, as that involves event classification and event relations, which is beyond the scope of this problem. This task is more geared towards establishing events as a concept that can be annotated in Hindi. We tackle the challenge of event detection from the perspective of maximum capture of event information given the nuances of Hindi grammar.

This task is daunting for a language like Hindi, because its syntactic properties (such as *karakas* and verb complements) tend to be governed by the semantics of the sentence, which is further emphasized by the relatively free word order. Therefore, the combination of our schema and guidelines explains the possible sentential structures in which an event can occur, accounting for characteristics such as fragmentation, stative constructions and negation.

2 Related Work

Automated event detection as a problem has been studied for a long time. A fundamental issue in solving this problem lies in how we define events in a given context (Goyal et al., 2013). To the best of our knowledge, there has been no prior work on a semantico-syntactic definition of events in Hindi. There have been multiple attempts to define events in English, but due to the varied complexities in definitions of events, most of the research focuses on limited guidelines for event detection in English texts like ACE (Doddington et al., 2004), TimeML (Pustejovsky et al., 2003), RED (O'Gorman et al., 2016), ECB (Bejan and Harabagiu, 2010), and ERE (Song et al., 2015). TimeML, a specification language for events and temporal expressions occurring in English texts, was developed to enhance natural language question answering systems about events and entities in news articles as a part of the TARSQI (Temporal Awareness and Reasoning systems for QA) Project[2]. TimeML guidelines have been adapted to languages such as Korean (Im et al., 2009), Italian (**?**), French (Bittar et al., 2011a) and Persian (Yaghoobzadeh et al., 2012). Hindi event detection was motivated and inspired by TimeML (Pustejovsky et al., 2003),

All the above mentioned guidelines were developed in order to structure the data for a particular system, context or purpose. ECB was developed for identifying coreference in news articles; ACE for the purpose of automatic inference of entities, relations and events from data. ERE was developed as a lighter-weight version of ACE with the goal of making annotation more convenient by consolidating some of the annotation type distinctions that were found to be the most problematic in ACE, as well as removing some more complex annotation features, and making the annotation more consistent across annotators.

ISO-TimeML (Pustejovsky et al., 2010) presents an international standard markup language for the annotation of events using the `<EVENT>` tag, temporal expressions using the `<TIMEX3>` tag, and links between entities such as temporal links `<TLINK>`, aspectual links `<ALINK>` and modal subordination `<SLINK>`. Multiple languages have datasets annotated based on ISO-TimeML including French (Bittar et al., 2011b) and Italian (Caselli et al., 2011). Our work deviates from ISO-TimeML in that we only define events for Hindi in this paper. The motivation for this deviation is that the definition of events in both TimeML and ISO-TimeML are syntactic in nature, whereas our definitions have a semantic basis, leading to a change in the very nature of the annotation.

[2]http://www.timeml.org/tarsqi/index.html

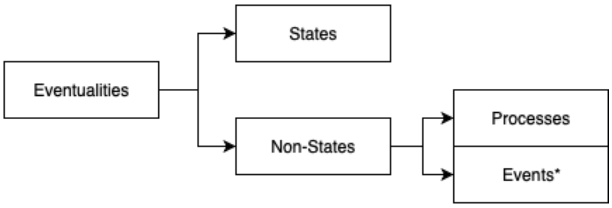

Figure 1: Bach's Eventualities

3 Introduction to Events and States

Defining the concept of events is deeply rooted in the linguistic study of verbs. In this section we lay the foundation for our definition of events. We also highlight the differences between states and non-states and propose our definition of events.

3.1 Eventualities

Initial works of Vendler (1957) on aspectual classification of verbs, classify verbs into four broad categories: states, accomplishments, achievements and activities, but it was the work of Bach (1986) that brought forth the notion of **eventualities**. He defined eventualities as aspectual phenomena and proposed that they were a "broader" notion of events. The idea was to include all durative phenomena which are telic and atelic, such that they could be classified as states, processes and events. The three components are defined using concepts of durativity (having a duration) and telicity (having an explicit condition of termination) (Moens and Steedman, 1988). Bach (1981, 1986) proposes the following division of 'eventualities'

1. **States** : Notions that are durative and changeless, e.g, hate, resemble
2. **Processes** : Notions that are durative and atelic, e.g, raining, sleeping
3. **Events** [3] : Notions that are momentaneous or telic, e.g, walk, slap, tap

3.2 States and Non-States

The categorization based on durativity and telicity captures the basic distinction between states and non-states. This distinction was de-emphasized in earlier works such that those of Vendler (Vendler, 1957). Bach's eventualities distinguish states from non-states (processes and actions) by recognizing the fact that there is no change involved in states and there is no explicit condition of termination.

Pustejovsky (1991) also followed this taxonomy of states, events and transitions, however, providing a more detailed understanding of subeventual structures in event types. This structure can only exist for non-states, given that any nesting in stative constructions will collapse to give a single description of the entity.

3.3 Events

Following the concept of eventualities as defined by Bach (Bach, 1986), we propose the definition of events as: **eventualities that are non-states and involve zero or more participants and attributes**. Essentially, events in Hindi are a combination of Bach's events and processes, and we exclude states from this definition entirely. This has been done for reasons explained in Section 3.4

Events are independent abstract concepts in the real world and not textual expressions bound to specific words or phrases. Events are lexicalized by languages as event mentions (Section 4.1). Given

[3] This refer's to Bach's distinction of eventualities into events. Our definition of events is introduced in Section 3.3

the unique syntactic constructions and semantic coverage associated with lexical items in different languages, each language captures event mentions differently. Events and event mentions can be better understood by looking at the example below :

1.

rameS	*ne*	*KAnA*	***KAyA***
Ramesh	(erg.)	food	ate

"Ramesh ate the food."

1.

usne	*apnA*	*kAm*	***kIyA***
He/She	(gen.)	work	did

"He/She did his/her work."

Here, the event is the notion of eating and doing respectively. The phrases *KAyA* and *kIyA* are **event mentions** that refers to the actual event of eating and doing respectively.

3.4 Deconstructing TimeML Events

Pustejovsky et al. (2003)'s definition of events, in TimeML, can be deconstructed into the following two distinct concepts.

- *We use event as a cover term for situations that happen, occur, hold, or take place* : These are captured as dynamic events (referred to henceforth as states), such as causing or intending to cause a change in the condition of involved entities.
- *We also consider as events those predicates describing states or circumstances in which something obtains or holds true*: These are captured as states, which define the condition of the involved entity at a point of time.

The second part of the TimeML event definition, concerned with states, causes trouble when applied to Hindi. In Hindi, addition of a verbalizer transforms almost all adjectives and nouns into events, however it is possible that the noun or adjective (without the verbalizer) does not actually represent an event. Consider the following example :

1.

Israel	*mein*	*gas*	*mask*	*kI*	***kamI***	*ke*	*kAran*	*unhe*	*AyAt*	*baDhAni*	*padi*
Israel	in	gas	mask	of	**shortage**	(gen.)	reason	they	import	increase	had-to

Due to a shortage of gas masks in Israel, they had to increase imports.

2.

Israel	*mein*	*gas*	*mask*	*kI*	***kamI***	***hone***	*ke*	*kAran*	*unhe*	*AyAt*	*baDhAni*
Israel	in	gas	mask	of	**shortage**	**to-be**	(gen.)	reason	they	import	increase
padi											
had-to											

Due to a shortage of gas masks in Israel, they had to increase imports.

In the first sentence, "*kamI*" denotes the state of shortage. While in the second sentence, we see that the verbalizer "*hone*" very distinctly makes the span "*kamI hone*" an event. However, semantically, the statements are equivalent, meaning that the existence of an event in the second implies the existence of an event in the first. Intuitively, and as observed while developing the annotation specifications, it was found that the annotation of states was challenging, due to their lack of syntactic uniformity. This made it difficult for manual annotators to confidently annotate states. To find the annotators' belief in the correctness in their annotation of an event, the *annotator confidence* parameter was introduced. If annotators felt uncertain about an annotation they could choose to give it a low confidence tag. By default every annotation is given a high confidence tag. It was found that states comprised an overwhelming majority of low confidence scores, making it challenging to use these annotations due to the high degree of uncertainty. Hence, for the remainder of this paper, the annotated data is focused *solely on events*.

4 Annotation Schema

In this section, we define the annotation schema for capturing the maximal amount of information in order to capture those textual mentions which are events. We define the terms event mention, event trigger and event core.

4.1 Event Mention

An event mention is the textual span expected to provide complete information of the event in terms of meaning, temporality and aspect. The event mention is therefore composed of four major parts, the event nugget (section 4.2), the tense marker, the aspect marker, and an optional polarity (negation) marker (section 4.4).

4.2 Event Nugget

Mitamura et al. (2015) defines the event nugget as ”a semantically meaningful unit that expresses the event in a sentence. An event nugget can be either a single word (main verb, noun, adjective, adverb) or a continuous or discontinuous multi-word phrase.”

Single word event nugget: That word in the event mention which contains the most semantic information about the event, and governs its argument structure is the event nugget. Semantically, it governs both the nature and the change in state of the participating entities.

2.	*rAkeS*	*uskA*	*KAnA*	[***KA***	*rahaa*	*hai*]
	Rakesh	his/her	food	eat	(prog.)	is

”Rakesh is eating his/her food”

In the example above, the word ***KA*** is the event nugget, because it is the only word in the event mention *KA rahA hai* that defines its argument structure. The nature of the event is governed by the meaning of *KA*.[4]

Multi-word event nugget: Here the event nugget is a multi-word phrase, that is usually non-compositional in meaning. In case of multi-word event nuggets, the semantics of the event is provided by all the words in the event nugget, therefore the nature of the participating entities are governed by the meaning of the phrase and therefore the change in state of the participants will also be different. For example:

3.	*rAkeS*	*uskA*	[sar	***KA***	*rahA*	*hai*]
	Rakesh	his/her	head	eat	(prog.)	is

”Rakesh is annoying him/her.”

The phrase *sar KAnA* literally means to annoy, which takes a set of arguments which are distinct from the nature and change in state of the entities from the event *KAnA* (to eat).

4.3 Fragmented Events

Hindi is a relatively free word order language. A consequence of that is the fragmentation of events which splits the event mention across a sentence, as a result of which the insertion of a phrase (or clause) within the event mention is not ungrammatical. An example of a fragmented event:

4.	*rAm*	*ne*	***nirdeS***	*gusse*	*se*	***diyA***
	Ram	(erg.)	order	anger	with	gave

”Ram gave the order angrily.”

[4] The event nugget determines that among the participating entities, one has the ability to eat and the other, to be eaten. This borrows from the Paninian framework of *yogyatA* or semantic expectation (Bharati and Sangal, 1993).

4.4 Capturing Negation

Events with a negative polarity in Hindi are temporally bound. They represent an instance where an event does not occur for a period of time. The positive event can be expected to occur otherwise. The concept of temporal bound of events with a negative polarity is introduced by explicit negative markers such as *nahI*, *nA* and a few more. To account for events with a negative polarity, we mark polarity indicators in the event mention.

5.

rAm	*ne*	*KAnA*	***nahI***	***KAyA***
Ram	(erg.)	food	not	ate

"Ram didn't eat the food."

5 Annotation Guidelines

While detecting events in Hindi, there are various syntactic features that need to be accounted for. These syntactic considerations are only made while annotating the span of an event. Eventiveness of a textual unit is independent of the syntax. As Hindi is mostly an analytic language, verbs in Hindi have tense and aspect markers as individual lexemes. This, along with the presence of phrasal conjuncts and conjunct verbs (Begum et al., 2011a), results in ambiguity in detection of event mentions as shown below.

5.1 Verbs and Verb Complexes

Verbs are parts of speech that denote action. Therefore, by definition, verbs and verbal predicates are events. Verbs in Hindi can be simple verbs, compound verbs, conjunct verbs and phrasal verbs. Unlike most languages, Hindi has the property of verbal constructions that span multiple words. This is due to the analytical nature of the language, and leads to constructions which can be modeled both semantically or syntactically.

Verbs in Hindi have distinct tense and aspect markers. Furthermore, nouns are verbalized by using a verbalizer, a separate lexical item which is independently a verb, but which in the presence of a noun transforms it into a verb with the same semantic implications as the noun. The list of verbalizers is a closed class of words, but allowing such a construction leads to multiple idiomatic and phrasal constructions.

The syntactic classification of these phenomena provides two basic classifications of verbs: predicate constructions and light verb constructions. However, these classifications do not show the relation of the noun to the nature of the verb, and are therefore not enough to explore the eventive nature of the verbal constructions. Therefore, we use a more semantic approach, subdividing predicates into simple verbs and compound verbs (without and with aspectual constructions respectively), and syntactic and phrasal conjuncts (non-idiomatic and idiomatic constructions in the presence of nouns/adjectives with the verbs). Conjunct verbs, due to their semantic implications in Hindi have been detailed separately in subsection 5.2.

- **Simple verbs** : are event mentions constructed by using only the verb, or using the verb and a tense marker (verb + inflectional "*tA*"). Simple verbs denote the habitual nature of the event or action. An example of this construction is:

  6.

rAm	*roz*	*shAm*	*ko*	*ghar*	***jAtA***	***hai***
Ram	daily	evening	(acc.)	home	goes	is

  "Ram goes home everyday in the evening"

- **Verbs with auxiliaries**: Verbal auxiliaries include aspect markers and modal indicators that give more information about the event itself. These can be *inchocative predicates* (predicates that denote coming to existence of a situation), *aspectual constructions* or simply denoting the possibility of an event. These may also occur in combination, such as *modal aspectual construction*.[5]

7.

SIlA	*kAm*	*par*	***jA***	***rahi***	***hai***
Sheela	work	to	go	cont-fem	is

"Sheela is going to work."

5.2 Conjunct Verbs and Phrasal Conjuncts

In Hindi, a conjunct verb is a complex predicate of the construction "Noun/Adjective + Verb" (Begum et al., 2011a). This construction raises the pertinent question of when a noun should be included in the event span while annotating the verbal event. Nouns, when occurring without a postposition with a verb, occur in one of the following contexts: as a participant, as a conjunct verb, and as part of an idiom or phrase.

- **Conjunct Verbs**

  In Hindi, a noun can be "verbalized" by addition of a verbalizer to denote action. The resultant multi-word expression, will behave exactly like a verb, with an argument structure, and temporal and aspectual characteristics of the verb, but the semantic characteristics and meaning of the noun.

  In the example below, *madad* (help), a noun combines with the verb *kI* (to do) to form the event core:

  8.

rAma	*ne*	*SyAma*	*kI*	***madad***	***kI***
Ram	(erg.)	Shyam	(gen.)	help	did

  "Ram helped Shyam."

- **Phrasal Conjuncts**

  An event mention as a phrasal conjunct includes a noun, and a verb which is *not* a verbalizer. Both the noun and the verb combine to form the event nugget. While the concept being represented by the phrasal event is realized by embedding the noun, the verbalizer presents the concept as an action. In the example below *paisA KAyA* is an idiom in Hindi which, although literally translates to "eating money", is usually interpreted as "scamming someone of their money".

  Unlike conjunct verbs, the semantic information in a phrasal conjunct can not be isolated to a single lexical item, rather, it is a combination of the noun and the verb that provide the meaning of the phrase.

  9.

rAma	*ne*	*SyAma*	*ka*	***paisA***	***KAyA***
Ram	(erg.)	Shyam	(gen.)	money	ate

  "Ram scammed Shyam"

Contemporary literature analyses similar events as light verb constructions in languages like Hindi and Persian, among others (Vaidya et al., 2016), from a syntactic perspective. While the surface structure is important for the annotation of phrasal events, the difference in nomenclature arises from the fact that phrasal events are not only syntactically unique to their conjunct form (Begum et al., 2011b), but also because the task of event annotation involves extracting information about the semantics of the conjuncts, which is not directly implied or studied in light verb constructions. Therefore, in this framework, they have been referred to as phrasal events rather than light verbs.

[5] Some linguists consider the suffix *tA* as an inflectional perfective aspect marker. However that debate does not concern our annotation of the event span. From the perspective of lexical annotation, verbal auxiliaries are included in the event span.

5.3 Nouns

Nouns categorized as events are either those, which are indicative of an action, or those which can be associated to the aspectual or temporal representation of another event. Nouns classified as events are either those that carry a predicate structure as a result of their nominalization or those whose inherent sense is eventive. For named events, we will be annotating the whole name as the event. This is denoted by the existence of all the above markers (participant or setting) and an attached NP that identifies it as a unique occurrence. A noun which is referred to by an event after it, along with an indication of temporality also indicates that the noun is an event.

10.	[***BUk***	***haDtAl***]	*ke*	*daurAn*	*annA*	*hazAre*	*ji*	***bImAr***	***paD***	***gaye***
	Hunger	strike	(gen.)	during	Anna	Hazare	sir	sick	fall	went

During the hunger strike Anna Hazare sir fell sick.

The noun phrase almost distinctly denotes the participation of an entity or the spatio-temporal setting of the occurrence of that event. This can be used to identify the trigger of an event, and hence denote it. A noun is considered to be an event if it occurs in one of the two given contexts:

1. Nouns followed by temporal or time related expressions suggesting that they can be used as (or are) referential in nature for other events.
2. Nouns that are indicative of having lasted a period of time. The duration can be in prolonged or can be immediate.

6 Challenges in Guideline Development

For developing our guidelines, we followed an iterative and incremental approach over three phases. In each phase, a version of the guidelines would be drafted that would then be validated by annotating events in accordance with the current version and then manually investigating the annotations to account for any problems or shortcomings. Shortcomings were majorly identified by calculating inter-annotator agreement and analyzing conflicting annotations. Inter-annotator agreement (strict match) was calculated using the Fleiss Kappa score (McHugh, 2012). Once these flaws were detected, the guidelines would then be accommodated to account for them, creating a new version. The cycle would continue until we reached an exhaustive set of guidelines.

Each version of the guidelines was tested on 50 unique news articles from the Hindi newspaper Dainik Jagran[6]. In each phase, the 50 articles were annotated by 4 annotators, all of whom were native Hindi speakers and had basic training in the field of linguistics. All annotation were done using the BRAT annotation tool Stenetorp et al. (2012a)

6.1 Phase 1

- **Draft**: One of the major challenges we faced in creating the first draft of the guidelines was finalizing the definition of events. As our research is focused on open-domain event detection, our preliminary definition of events was inspired by that of TimeML Pustejovsky et al. (2003). At the end of Phase 1, we had version 0.1 of the guidelines. 50 articles were then annotated as per these guidelines.
- **Challenges**: After manually analyzing the annotations, it was detected that copulative constructions contributed to most of the conflicting annotations. We also observed that annotation of event spans was very inconsistent between annotators. Fleiss' Kappa score observed: 0.59.

[6]https://jagran.com

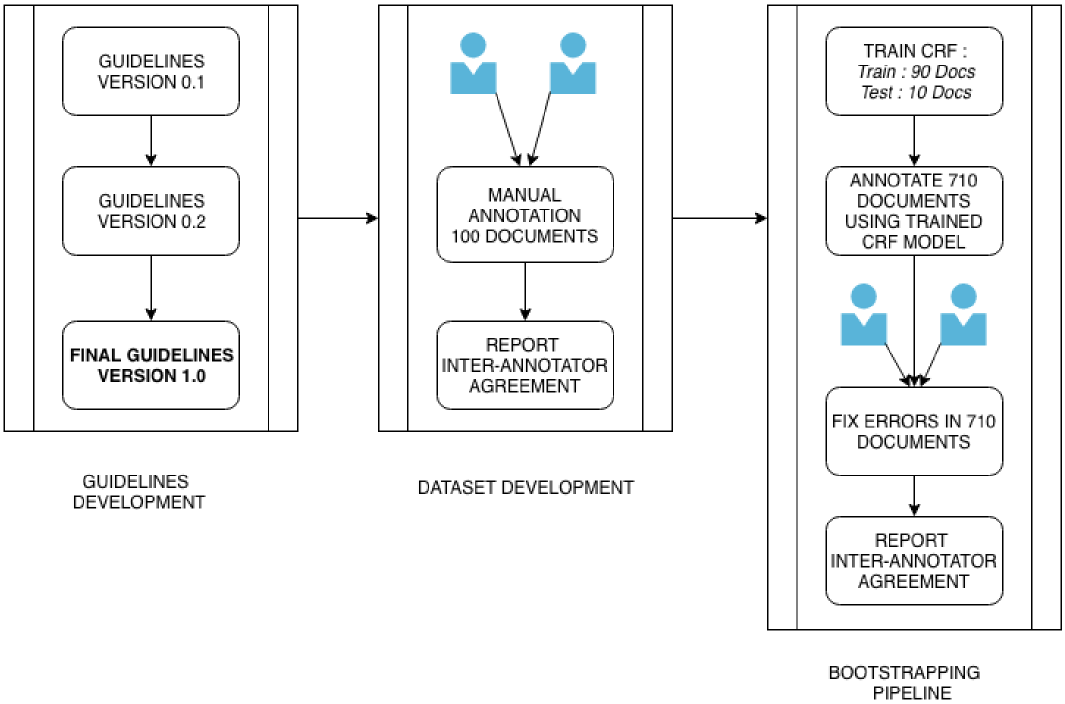

Figure 2: Dataset Development Pipeline

6.2 Phase 2

- **Draft**: During Phase 2, the guidelines were modified to exclude copular constructions from being annotated as events. This was done because copulative constructions do not always highlight a change in the properties of the direct object of the copulative verb and hence. To account for inconsistencies in event span annotation, the guidelines now incorporated rules to mark the tense, aspect and modality (TAM) information of the event within the span of the event (Section 4.2). 50 articles were then annotated as per these guidelines.

- **Challenges**: Manual evaluation of annotations during this phase highlighted three major issues. The first was relation to conflicting annotation of idiomatic expressions (example 1 below). The second being that some annotators had marked the direct object of the event trigger in the event span while the other annotators had not. The final issue showed that Hindi captures certain events (a single semantic idea) in non-contiguous (fragmented) syntactic sequences (example 2 below). Most annotators only marked a part of such constructions. Fleiss' Kappa score observed : 0.72. Examples of conflicting annotations:

11.

rAm	*se*	*dUr*	*raho,*	*vaha*	***maKan***	***lagAne***	*mein*	*ustAd*	*hai*
Ram	from	away	stay,	he/she	butter	putting	in	expert	is

"Stay away from Ram, he's an expert at flattering people."

12.

unhone	***ghUs***	*parson*	*subah*	***dI***
They + (erg.)	bribe	day-before	morning	gave

"They gave the bribe day before yesterday morning."

6.3 Phase 3

- **Draft**: During Phase 3, the guidelines were further modified to account for the following changes. Idiomatic expressions would be marked events because they derive meaning from ontological roots rather than constructional syntactic uniformity (Butt, 2010). Rules were established to account for annotation of fragmented events (Section 4.3). We now also accounted for events that were and

Data Point	Value
Number of Articles	810
Number of Sentences	13949
Number of Tokens	242201
Number of events	20190
Average Sentence Length (Words)	18

Table 1: Dataset Metrics

Element	Train	Test	Validate
Sent	10318	2430	1201
Tokens	175967	44143	22091
Events	15257	3316	1617
Verbal Events	13808	3061	1480
Nominal Events	1449	255	137

Table 2: Data Split

Features
Word Identity (WI)
Part-of-Speech (POS)
Bi-gram and tri-gram features
Beginning Of Sentence (BOS)
Window features: POS, WI

Table 3: CRF Features

were not syntactically bound with an embedded nominal (Section 5.2). This was the final phase of the guideline development cycle and resulted in Version 1.0 of the Hindi Event Annotation Guidelines. 50 articles were then annotated as per these guidelines.

- **Challenges**: Manual evaluation of annotation during this phase revealed minor discrepancies between annotators. Most of these discrepancies were regarding a missed/skipped annotations. Fleiss' Kappa score observed: 0.86.

7 Dataset

For the task of automated event detection in Hindi, we introduce a gold standard event annotated corpus. This *gold standard dataset*, comprises of 810 event annotated news articles from the financial domain of the Hindi newspaper, Dainik Jagran. The articles have been extracted date from July-December, 2017. The articles in the dataset were manually curated and selected to account for multiple types of events in varying syntactic and semantic conditions. The metrics of the gold standard dataset are described in Table 1. The data was annotated using the BRAT (Stenetorp et al., 2012b) annotation tool.

7.1 Bootstrapping Dataset Development

Once the guidelines were finalized (Version 1.0), four annotators annotated 100 news articles (D1) in accordance with them. We observed a strong inter-annotator agreement, with a Cohen's Kappa score (Cohen, 1960) of 0.84 (strict match) which re-affirmed our confidence in the clarity and coverage of the guidelines.

To further expand our dataset, we use a bootstrapping approach defined in figure 2. Using 90 out of the 100 articles from D1, we train a linear chain CRF to predict events in Hindi text. Features used to train the CRF are shared in table 3. The CRF is then tested on the remaining 10 articles and reports an F1 score of 65.22. We used the trained CRF to annotate the another 710 news articles (D2). Given the low F1 score of the CRF, we were aware that the annotations were erroneous. The same team of annotators that worked on D1, now manually reviewed and resolved all errors in the annotations done by the CRF. The annotators showed a final inter-annotator agreement of 0.79. At the end of this we had 810 articles annotated for events in Hindi (D3), which formed our gold standard dataset. The dataset will be made publicly available on an easily accessible platform upon validation by the community.

8 Conclusion and Future Work

In this paper, the concept of event detection is introduced for Hindi text along with a comprehensive set of annotation guidelines and specifications for detecting events in Hindi text. We introduce an event annotated dataset for Hindi news articles, which is the first dataset in an Indo-Aryan language of this kind to the best of our knowledge. The guidelines presented allowed the annotation of the 810 article dataset with high agreement among annotators, indicating the robustness of the annotation scheme.

This paper is an attempt to preliminarily establish this new direction of NLP research in Hindi. This task of event detection can now be introduced to other Indo-Aryan languages with ease, including Urdu, Bengali, Punjabi, Marathi, Oriya and so on, with minimal changes to the event detection guidelines. These guidelines can therefore be applied for event mention annotation over a family of languages which are low-resource in nature.

The dataset introduced can be used for further tasks such as stative event detection, event classification, and annotating event-entity and event-event relations. This information can be used for tasks like factoid question answering, extractive summarization and other related tasks.

The ISO-TimeML annotation mechanism can be adopted in order to create a TimeBank for Hindi as has been done for the other languages mentioned above, but that is beyond the scope of this paper.

References

Bach, E. (1981). On time, tense, and aspect: An essay in english metaphysics.

Bach, E. (1986). The algebra of events. *Linguistics and philosophy 9*(1), 5–16.

Begum, R., K. Jindal, A. Jain, S. Husain, and D. M. Sharma (2011a). Identification of conjunct verbs in hindi and its effect on parsing accuracy. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pp. 29–40. Springer.

Begum, R., K. Jindal, A. Jain, S. Husain, and D. M. Sharma (2011b). Identification of conjunct verbs in hindi and its effect on parsing accuracy. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pp. 29–40. Springer.

Bejan, C. A. and S. Harabagiu (2010). Unsupervised event coreference resolution with rich linguistic features. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 1412–1422. Association for Computational Linguistics.

Bharati, A. and R. Sangal (1993). Parsing free word order languages in the paninian framework. In *Proceedings of the 31st annual meeting on Association for Computational Linguistics*, pp. 105–111. Association for Computational Linguistics.

Bittar, A., P. Amsili, P. Denis, and L. Danlos (2011a). French timebank: an iso-timeml annotated reference corpus. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pp. 130–134. Association for Computational Linguistics.

Bittar, A., P. Amsili, P. Denis, and L. Danlos (2011b). French timebank: an iso-timeml annotated reference corpus. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pp. 130–134. Association for Computational Linguistics.

Butt, M. (2010). The light verb jungle: Still hacking away. *Complex predicates in cross-linguistic perspective*, 48–78.

Caselli, T., V. B. Lenzi, R. Sprugnoli, E. Pianta, and I. Prodanof (2011). Annotating events, temporal expressions and relations in italian: the it-timeml experience for the ita-timebank. In *Proceedings of the 5th Linguistic Annotation Workshop*, pp. 143–151. Association for Computational Linguistics.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and psychological measurement 20*(1), 37–46.

Doddington, G. R., A. Mitchell, M. A. Przybocki, L. A. Ramshaw, S. Strassel, and R. M. Weischedel (2004). The automatic content extraction (ace) program-tasks, data, and evaluation. In *Proceedings of the 4th Language Resources and Evaluation Conference*.

Goyal, K., S. K. Jauhar, H. Li, M. Sachan, S. Srivastava, and E. Hovy (2013). A structured distributional semantic model for event co-reference. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Volume 2, pp. 467–473.

Im, S., H. You, H. Jang, S. Nam, and H. Shin (2009). Ktimeml: specification of temporal and event expressions in korean text. In *Proceedings of the 7th Workshop on Asian Language Resources*, pp. 115–122. Association for Computational Linguistics.

McHugh, M. L. (2012). Interrater reliability: the kappa statistic. *Biochemia medica: Biochemia medica 22*(3), 276–282.

Mitamura, T., Y. Yamakawa, S. Holm, Z. Song, A. Bies, S. Kulick, and S. Strassel (2015). Event nugget annotation: Processes and issues. In *Proceedings of the The 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pp. 66–76.

Moens, M. and M. Steedman (1988). Temporal ontology and temporal reference. *Computational linguistics 14*(2), 15–28.

O'Gorman, T., K. Wright-Bettner, and M. Palmer (2016). Richer event description: Integrating event coreference with temporal, causal and bridging annotation. In *Proceedings of the 2nd Workshop on Computing News Storylines (CNS 2016)*, pp. 47–56.

Pustejovsky, J. (1991). The syntax of event structure. *cognition 41*(1-3), 47–81.

Pustejovsky, J., J. M. Castano, R. Ingria, R. J. Gaizauskas, A. Setzer, G. Katz, and D. R. Radev (2003). Timeml: Robust specification of event and temporal expressions in text. *New directions in question answering 3*, 28–34.

Pustejovsky, J., K. Lee, H. Bunt, and L. Romary (2010). Iso-timeml: An international standard for semantic annotation. In *LREC*, Volume 10, pp. 394–397.

Song, Z., A. Bies, S. Strassel, T. Riese, J. Mott, J. Ellis, J. Wright, S. Kulick, N. Ryant, and X. Ma (2015). From light to rich ere: annotation of entities, relations, and events. In *Proceedings of the the 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pp. 89–98.

Stenetorp, P., S. Pyysalo, G. Topić, T. Ohta, S. Ananiadou, and J. Tsujii (2012a). Brat: a web-based tool for nlp-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 102–107. Association for Computational Linguistics.

Stenetorp, P., S. Pyysalo, G. Topić, T. Ohta, S. Ananiadou, and J. Tsujii (2012b). Brat: a web-based tool for nlp-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 102–107. Association for Computational Linguistics.

Vaidya, A., S. Agarwal, and M. Palmer (2016). Linguistic features for hindi light verb construction identification. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 1320–1329.

Vendler, Z. (1957). Verbs and times. *The philosophical review 66*(2), 143–160.

Yaghoobzadeh, Y., G. Ghassem-Sani, S. A. Mirroshandel, and M. Eshaghzadeh (2012). Iso-timeml event extraction in persian text. *Proceedings of COLING 2012*, 2931–2944.

Leveraging Multilingual Resources for Open-Domain Event Detection

Jaipal Singh Goud
IIIT Hyderabad
jaipal.singh@research.iiit.ac.in

Pranav Goel
IIIT Hyderabad
pranav.goel@research.iiit.ac.in

Allen J. Antony
IIIT Hyderabad
allen.antony@research.iiit.ac.in

Manish Shrivastava
IIIT Hyderabad
m.shrivastava@iiit.ac.in

Abstract

Natural Language Processing techniques for event detection based on deep learning rely on large collected data sets. As a result, there is little to no progress in the development of solutions for event detection in low resource languages. Driven by this motivation, our work attempts to introduce an architecture, the Multilingual Sequence Tagger (M-LiST) capable of training on a combination of four monolingual datasets and attaining state of the art performance for open domain event detection in three languages of the TempEval2 (Pustejovsky et al., 2010) corpus.

1 Introduction

Neoteric advances in neural architectures for sequence labeling tasks have managed to achieve impressive performances. Bi-directional Recurrent Neural networks with a CRF layer are the most popular method for sequence labelling using Deep Networks (Ma and Hovy, 2016; Lample et al., 2016; Athavale et al., 2016). Credit for the efficiency of these methods is in-part due to the availability of large-scale annotated datasets for a few languages, though such benefits of large annotated corpora are not enjoyed by most other languages. Developing annotated resources for all languages is both an expensive and significantly time-intensive task. Our work attempts to tackle this problem of event detection in languages with small-sized event corpora by proposing a neural Multi Lingual Sequence Tagger (M-LiST).

We treat this task as a *multi-lingual transfer learning* (MLTL) language adaptation task and attempt to learn language invariant features (Yarowsky et al., 2001). A major challenge in language adaption tasks is the difference in vocabulary of the source and target languages. Most multi lingual transfer learning tasks until now, have depended on the use of a parallel corpora (Täckström et al., 2013; Duong et al., 2013), bilingual dictionaries (Xie et al., 2018; Fang and Cohn, Fang and Cohn) or learnt language specific features. M-LiST attains independence from these language specific features by using multilingual word embeddings that are aligned in an adversarial manner (Conneau, Lample, Ranzato, Denoyer, and Jgou, Conneau et al.) without the use of any parallel corpus or bilingual dictionary.

M-LiST, ideally, attempts to learn a model with language-agnostic representations for sequence labeling via adversarial training on data from multiple source languages together. The **single trained model** can then be used for making predictions on data from **any of the source languages** without adapting any language specific features. In a special setting, an M-LiST model trained on multiple languages($L_1....L_n$) can also be used for making predictions on a new, unseen language (L_k) if L_k is similar to one of the languages the model is trained on.

The main contributions of the paper can be summarized are : (i) We propose a novel neural approach for event detection, capable of training on multiple datasets from different languages. (ii) We extensively evaluate the performance of our model's performance on data from 3 languages.

The paper is organized as follows : Section (2) highlights related work done on open domain event detection. Section (3) breaks down M-LiST and explains the various components it's composed of. Section (4) explains in detail the training setup followed Section (5) that summarizes our experimental setup. Section (6) discusses our results and observations and Section (7) concludes our work.

2 Background and Related Work

Automated event detection, both in the open and closed domain is a well researched problem. The current best model for open domain event detection in Spanish (TempEval-2 dataset) was given by Wonsever et al. (2012). They experiment with two machine learning models for event recognition, viz., Conditional Random Fields (CRF) and an adapted version of a Support Vector Machine (SVM). A 70-30 train-test split was done on the dataset. In both cases, they used morphosyntactic attributes for event detection, some of which were sourced from the Freeling (Atserias et al., 2006) tagger (features included : token information, lemma, POS-tag, number, mood and tense), and others associated with word structure (capital letters, last four letters). To train their CRF, they used a window size of [-2,2] centered around the word they wanted to classify. They manage to achieve an impressive F1 score of 80.3.

Arnulphy et al. (2015) have the best model for English as well as French. They experiment with three machine learning based systems, viz., CRF, Decision Trees and an ensemble model of CRF and K-Nearest Neighbors. The common features used for both French and English include word-form, lemma and part-of-speech tag. For English, additional feature used was an tag for each word which indicated whether it belongs to one of the eight classes of synsets concerned with actions or events. Similarly, for French, a feature indicated for each word whether it belongs to the VerbAction (Tanguy and Hathout, 2002) and The Alternative Noun Lexicon (Bittar, 2010) lexicons or not. They achieve an F1 score of 86.0 for event detection in English and an F1 score of 83.0 for French.

3 M-LiST Architecture

This section defines the various components that M-LiST is composed of.

Word-Alignment We have adopted the unsupervised method proposed by Conneau, Lample, Ranzato, Denoyer, and Jgou (Conneau et al.) for aligning monolingual word embedding spaces of each language. For our experiments, we have used word embeddings from fastText (Bojanowski et al., 2017). We treat English as the target embedding space and align all the source language spaces (French and Spanish) to this space.

Feature Encoder : In this task, a document d is made up words $w_1, w_2, \ldots w_n$. We encode each word, w_i, using two embeddings - a character level embedding and the word's aligned embedding. The character level representation of a word is generated with the help of a character-level CNN. Further, both of these embeddings are concatenated to make a final embedding for the word. The embeddings for all the words are passed sequentially into an Bi-directional LSTM network which at each time step outputs a hidden state h_i, for every input word w_i. Each hidden state h_i is then passed into a language discriminator (LD) and a Sequence Tagger (ST).

Language Discriminator : The Langauge Discriminator (LD), which is made up of a Gradient Reversal Layer (GRL) and two Fully Connected Layers and a Softmax Layer, tries to predict the language the word

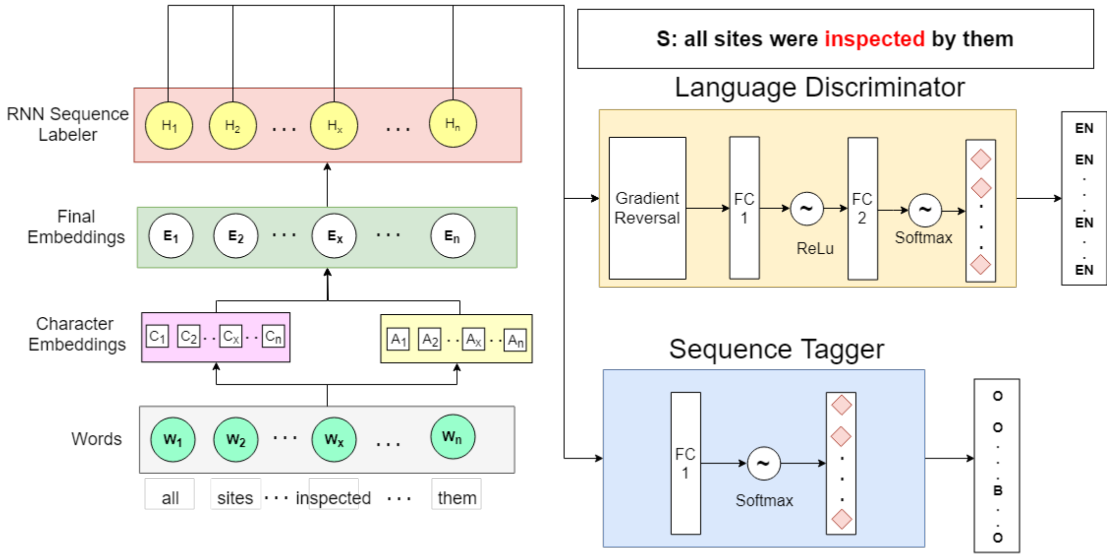

Figure 1: Architecture of the proposed model. The input sentence S is on the top right. The event in the sentence is highlighted in red color. The language discriminator tries to output the language of the input sentence, which in this case is EN(English).

w_i belongs to.

Sequence Tagger : The Sequence Tagger (ST) made up of a Fully Connected Layer followed by a Softmax Layer, attempts to assign a task label to the word w_i using h_i. The softmax function ensures that the network outputs are normalized between zero and one, and that they sum to one at every timestep. This means they can be interpreted as the posterior probabilities of the tag at a given word.

Joint Training : During training via backpropogation, the gradient reversal layer in the LD multiplies the gradient by a certain negative constant. While training, the learning objective is to minimizes the label prediction loss and maximize the language classification loss. Gradient reversal ensures that the feature distributions over the languages are made similar, making it difficult for the language discriminator to detect the language and, thus resulting in the langauge-invariant features.

4 Training Setup

To train M-LiST, we follow a setup that is trained on labeled documents $D = \{d_1, d_2, \ldots d_n\}$. Each document d_h is sourced from a language $L_i \in L$ where $L = \{L_1, L_2, \ldots L_k\}$ is a set of languages and each document d_h has associated with it a set of target labels $y_h = \{y_{h_1}, y_{h_2}, \ldots y_{h_m}\}$ where $m = \|d_h\|$. Also associated with each document is a language identifier label z_i. Our goal is to train a classifier capable of predicting the task label set y_i for every document, independent of the language identifier label z_i. In other words, we want to have a function f such that $y_1 = f(d_i)$ is independent of z_i.

In order to achieve this, we model f as an encoder than transforms an input d_i into a hidden representation $h(d_i)$ and a classifier, c, that predicts y_i based on $h(d_i)$. Thus, our main task prediction, i.e., $y_i = c(h(d_i))$, is independent of z_i if $h(d_i)$ is independent of z_i.

Language	Tokens	Event Mentions
English	53000	5688
Spanish	68000	12385
French	13000	2100

Table 1: TempEval2 Training Data Statistics

4.1 Adversarial Training

We employ the adversarial training setup inspired by (Goodfellow et al., 2014; Ganin et al., 2016; Beutel et al., 2017). We use this setup in order to make f independent of z_i. During training, we introduce an adversarial classifier, ad, which aims to predict z for a given $h(d_i)$, while at the same time, the encoder tries to fool the adversarial classifier. As a result, the setup tries to optimise the below functions in a joint fashion.

$$arg \max_{ad} \; L(ad(h(d_i)), z_i)$$
$$arg \min_{h,c} \; L(c(h(d_i)), y_i) - L(ad(h(d_i)), z_i)$$

where $L(x, x')$ is the loss function. As a result, the setup gets trained to form $h(d_i)$ such that it is minimally influenced by the language label z_i while at the same time influenced maximally by the task at hand, i.e., y_i. In practice, this is achieved by having a Gradient Reversal Layer (GRL) (Ganin et al., 2016). The GRL is a layer between the encoder and the adversarial classifier ad.

During the forward pass of the training, the GRL, which is represened by g_λ acts as an identity, while in the backward pass of the training, it scales the gradients passed onto it by a factor of $-\lambda$. This results in the encoder to receive reverse gradients and thus the equation to optimise becomes:

$$arg \min_{h,c,ad} \; L(c(h(d_i)), y_i) + L(ad(g_\lambda(h(d_i))), z_i)$$

5 Experiments

For our task of multi-lingual open-domain event detection, we use the TempEval2 shared task (Pustejovsky et al., 2010) corpus annotated in accordance to the TimeML (Pustejovsky et al., 2003) guidelines. The data statistics have been provided in Table 1. TimeML was first developed in 2002 and was extended in TERQAS (Time and Event Recognition for Question Answering Systems) workshop, which focused on the issue of answering temporally based questions regarding events and entities in news articles. We use the data for 3 different languages viz. English, French and Spanish from the TempEval2 dataset.

5.1 Experimental Setup

Our task is to identify meaningful events in text. The TempEval2 dataset has event annotated data for Italian, English, French and Spanish. The data is annotated in the IOB format. At a word level, 'B' represents the first token of an event, 'I' represents all the other tokens of an event and O represents the tokens which are not a part of any event in the sentence. During training, the model is trained in a joint manner. We use the Adam optimizer (Kingma and Ba, 2014) in all the variations of our model. The model is given an annotated sentence and the language the sentence belongs to. This is done for every sentence of the 3 languages.

The character embeddings and the aligned embeddings are of length 300 each. The hidden representations h_i given by the Bi-LSTM is of size 300. We use hidden state size as 100 for the forward and backward

Method	English			Spanish			French		
	P	R	F_1	P	R	F_1	P	R	F_1
Current Best	86.0	86.0	86.0	**84.7**	76.4	80.3	87.0	79.0	83.0
M-LiST	**86.2**	**86.7**	**86.4**	84.4	**87.5**	**85.9**	**96.5**	**94.0**	**95.3**

Table 2: Performance of M-LiST on the TimeML Dataset

LSTM respectively in the Bi-LSTM, and size 200 for the LSTM decoder.We use dropout with rate 0.5 on both the input and output of all LSTMs. The mini-batch size is set to 1. The number of training epochs are limited to 200. For the word embeddings, we use aligned embeddings and character lever embeddings. The method to obtain aligned embeddings is described in Section 3. The other is a word representation generated based on its composed characters. We adopt a CNN onto the randomly initialized character embeddings, with 30 filters and filter window size 3. We use the negative log likelihood loss function.

6 Results and Observations

Prior approaches to detect events in text have heavily relied on using various language specific features such as POS, lexicon feature, ERW values & more. Our single model trained on multiple languages without the use of any language specific features achieves a better performance than all prior experiments on the same task in Spanish, English and French (Section 2) as shown in Table 2.

The results of our experiments reaffirm our confidence in M-LiST to learn patterns and concepts (events) common across languages. Prior to explaining why M-LiST performs significantly better on Spanish and French datasets, it's important to understand about the nature of our data. Event annotated corpora for French is relatively small in size while the one for Spanish is the largest of all. Despite being two separate languages, they both belong to the Romance branch of the Indo-European language family and share a similar syntactic structure of capturing event mentions. Legacy approaches for event detection have always relied solely on using monolingual datasets to automate event detection. This entails that older models are only good at predicting events that occur in a setting similar to their training data. This assumption leads us to believe that while experimenting with the French dataset, there is not enough representation in the training set to enable the model to learn the all possible ways in which events are captured in French. M-LiST solves this problem and enables us to leverage learnings from multiple languages (Spanish and English) and leverage them to provide enhanced results even when working with a small sized language such as French.

7 Conclusion

In this paper, we explored the task of event-detection by leveraging event annotated data from multiple languages. Our architecture achieves state of the art performance on three out of the four languages that are a part of the tempeval2 dataset. In the future, we plan to explore the applications of this architecture on other sequence labeling tasks and also it's ability to learn on datasets from more languages. We feel that interesting work can also be done in the domain of event detection by grouping together similar languages and experimenting with transfer learning models on them.

References

Arnulphy, B., V. Claveau, X. Tannier, and A. Vilnat (2015). Supervised machine learning techniques to detect timeml events in french and english. In *International Conference on Applications of Natural Language to*

Information Systems, pp. 19–32. Springer.

Athavale, V., S. Bharadwaj, M. Pamecha, A. Prabhu, and M. Shrivastava (2016). Towards deep learning in hindi ner: An approach to tackle the labelled data scarcity. In *13th International Conference on Natural Language Processing*, pp. 154.

Atserias, J., B. Casas, E. Comelles, M. González, L. Padró, and M. Padró (2006). Freeling 1.3: Syntactic and semantic services in an open-source nlp library. In *LREC*, Volume 6, pp. 48–55.

Beutel, A., J. Chen, Z. Zhao, and E. H. Chi (2017). Data decisions and theoretical implications when adversarially learning fair representations. *arXiv preprint 1707.00075.*

Bittar, A. (2010). *Building a TimeBank for French: a reference corpus annotated according to the ISO-TimeML standard.* Ph. D. thesis, Paris 7.

Bojanowski, P., E. Grave, A. Joulin, and T. Mikolov (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics 5*, 135–146.

Conneau, A., G. Lample, M. Ranzato, L. Denoyer, and H. Jgou. Word translation without parallel data. *arXiv preprint 1710.04087.*

Duong, L., P. Cook, S. Bird, and P. Pecina (2013). Simpler unsupervised pos tagging with bilingual projections. In *ACL.*

Fang, M. and T. Cohn. Model transfer for tagging low-resource languages using a bilingual dictionary. *arXiv preprint 1705.00424.*

Ganin, Y., E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky (2016). Domain-adversarial training of neural networks. *The Journal of Machine Learning Research 17*(1), 2096–2030.

Goodfellow, I., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio (2014). Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680.

Kingma, D. P. and J. Ba (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980.*

Lample, G., M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer (2016). Neural architectures for named entity recognition. In *Proceedings of NAACL-HLT*, pp. 260–270.

Ma, X. and E. Hovy (2016). End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Volume 1, pp. 1064–1074.

Pustejovsky, J., J. M. Castano, R. Ingria, R. Sauri, R. J. Gaizauskas, A. Setzer, G. Katz, and D. R. Radev (2003). Timeml: Robust specification of event and temporal expressions in text. *New directions in question answering 3*, 28–34.

Pustejovsky, J., M. Verhagen, X. Nianwen, R. Gaizauskas, M. Hepple, F. Schilder, G. Katz, E. Saquete, T. Caselli, et al. (2010). Tempeval2: Evaluating events, time expressions and temporal relations.

Täckström, O., D. Das, S. Petrov, R. McDonald, and J. Nivre (2013). Token and type constraints for cross-lingual part-of-speech tagging. *Transactions of the Association for Computational Linguistics 1*, 1–12.

Tanguy, L. and N. Hathout (2002). Webaffix: un outil d'acquisition morphologique dérivationnelle à partir du web. In *9e Conférence Annuelle sur le Traitement Automatique des Langues Naturelles (TALN-2002)*, pp. 254.

Wonsever, D., A. Rosá, M. Malcuori, G. Moncecchi, and A. Descoins (2012). Event annotation schemes and event recognition in spanish texts. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pp. 206–218. Springer.

Xie, J., Z. Yang, G. Neubig, N. A. Smith, and J. Carbonell (2018). Neural cross-lingual named entity recognition with minimal resources. *arXiv preprint 1808.09861*.

Yarowsky, D., G. Ngai, and R. Wicentowski (2001). Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of the First International Conference on Human Language Technology Research*, HLT '01, Stroudsburg, PA, USA, pp. 1–8. Association for Computational Linguistics.

An evaluation study of annotating semantic frames on top of syntactic dependencies

Merle Pfau
Universität Hamburg
merle.pfau@web.de

Heike Zinsmeister
Universität Hamburg
heike.zinsmeister@uni-hamburg.de

Abstract

This short paper presents an evaluation study of manually annotating semantic PropBank frames on top of TIGER dependencies. The evaluation is restricted to a controlled setting by annotating 10 highly frequent German verbs realized as finite verbs in 199 sentences (thereby excluding passive voice). We evaluate inter-annotator agreement on (a) selecting an appropriate semantic sense from English PropBank, (b) assigning the same semantic roles to the arguments. Our main interest is investigating the linking of semantic roles on top of syntactic functions, which we evaluate quantitatively and qualitatively: The findings show that the core roles ARG0 and ARG1 are mostly annotated on top of dependency subject and object relations and that 68 % of the other roles' instances are aligned with modifiers. We also discuss five linguistic phenomena in which the TIGER dependency annotation does not provide a scaffolding for semantic role assignment.

1 Introduction

A core task of natural language understanding is to identify semantic relations in sentences: who did what to whom by which means etc. Each verb comes with one (or more than one) particular set of semantic roles, which are canonically realized in terms of syntactic arguments, such as, the agent of an action is canonically realized as the nominative subject of the sentence, the patient or theme as the accusative object. However, work on argument alternations (e.g. Levin, 1993) shows that the linking of semantic roles to syntax is not necessarily isomorphic, passive diathesis being a prominent example of a non-isomorphic mapping or stimulus-subject psych verbs (Dowty, 1991) like German *X ängstigt mich* 'X frightens me'.

In this short paper we present an evaluation study of manually annotating semantic PropBank frames (Palmer et al., 2005) on top of German TIGER dependencies (Brants et al., 2004; Seeker and Kuhn, 2012). The evaluation is restricted to a very controlled setting that excludes known discrepancies in the linking between syntax and semantics such as the mentioned ones related to passive voice and psych verb constructions. Our goal is to determine the variation in the mapping between syntactic dependencies and semantic roles in apparently straightforward cases. In the following, we will briefly present semantic annotation of German corpora. Then we will describe our evaluation study in detail. The variation in mapping of semantic roles on top of syntactic dependencies is summarized in a quantitative analysis. Finally, we will discuss five phenomena in which semantic role assignment is not fully supported by the syntactic dependency annotation corresponding with the TIGER scheme.

2 Semantic roles

Semantic roles date back to the ancient Indian scholar Panini and are one of the oldest grammatical models (Gildea and Jurafsky, 2002, p. 247). They were reintroduced to linguistic research in the second half of the 20th century. Fillmore (1968) established the deductive concept of deep roles that represent the set of abstract entities related to specific verbs. Fillmore (1977) introduced the related concept of semantic frames which, in contrast, builds on data-derived sets of event-specific roles.[1] This latter model was used to create the semantic database FrameNet (Baker et al., 1998). The semantic database PropBank (Palmer et al., 2005) combines the idea of deep roles with a generalized set of semantic roles (Dowty, 1991), numbering the core arguments from ARG0 to ARG4 (see table 1).

3 Related work

There are only a few German corpora annotated with semantic roles. Burchardt et al. (2006) created the German SALSA corpus by manually annotating sentences from the TIGER corpus with FrameNet-like roles on top of syntactic constituency annotations.[2] The authors discuss cross-lingual divergencies including differences in argument realization, which we will also address but concerning a different phenomenon. SALSA marks phenomena with limited compositionality like support verb constructions and idioms with extra labels, which we did not adopt for keeping our investigation simple. For SALSA, the authors report an average number of frames per predicate of 2.8. Inter-annotator agreement between two annotators was 85 % on frames and 86 % on roles.

For the CoNLL-2009 Shared Task corpus (Hajic et al., 2012),[3] the SALSA corpus was semi-automatically converted to simplified dependency annotations and semantic PropBank labels (Hajič et al., 2009, p. 12 f.). The corpus also includes 1,140 sentences from the Europarl Corpus of parliament statements (Koehn, 2005) which had been manually annotated according to SALSA guidelines (Padó and Lapata, 2005) and were also converted for CoNLL along with the SALSA data. In the shared task they were used as out-of-domain test data for the newspaper-based SALSA training data (cf. Hajič et al., 2009).

Mújdricza-Maydt et al. (2016) re-annotated about 3,500 instances of 275 verbal predicates from the CoNLL-2009 version of the SALSA corpus with GermaNet senses (Hamp and Feldweg, 1997) and a revised scheme of VerbNet roles (Kipper Schuler, 2005) which are more expressive than PropBank roles and more restricted than FrameNet frame elements: The SemAnno corpus[4] also includes 450 annotated instances of 30 verbal predicates from the Dortmund Chat Corpus (Beißwenger and Storrer, 2008). Hartmann et al. (2017) extended about 3,000 instances of the SemAnno CoNLL-2009 data by adding a layer of PropBank-style annotations: The SR3de dataset[5] strictly aligns FrameNet-style, VerbNet-style and PropBank-style annotations. The authors evaluated the different frameworks by training a semantic role labeling tool for German on the dataset. The baseline results confirmed the initial assumption that PropBank roles generalize best, VerbNet roles second and FrameNet roles suffer most from training data sparseness. The authors did not investigate the interaction of syntactic dependency structure and (PropBank) semantic relations which is in the focus of our study.

All the manual annotation efforts above made use of English lexical resources as support for the manual annotation of German data. We interpret this as a proof of concept that it is possible to apply an English role inventory to a German corpus on a larger scale.

[1] Ruppenhofer et al. (2016, p. 11) describe the partly inductive approach as follows: "The frames that make up the core of the project's work do not come out of nowhere. [...] Whether we start with an idea or start with a word, the core of the frame development process has always been looking at corpus attestations of a group of words that we believe to have some semantic overlap, and dividing these attestations into groups. Afterward, we combine the small groups into large enough groupings to make reasonable frames at which point we may (equivalently) call the words targets, lexical units, or frame-evoking elements."

[2] SALSA corpus: `http://www.coli.uni-saarland.de/projects/salsa/`. SALSA 2.0 consists of 24,184 sentences annotated with 37,697 frame-evoking elements and 66,486 frame elements.

[3] CoNLL-2009 task description and gold data:`http://ufal.mff.cuni.cz/conll2009-st/index.html`.

[4] SemAnno corpus: `https://www.cl.uni-heidelberg.de/projects/GNVN_semanno/index.mhtml`.

[5] SR3de dataset: `https://www.cl.uni-heidelberg.de/projects/SR3de/data.mhtml`.

4 Evaluation study

4.1 Setting and Data

We extracted sentences from the dependency conversion of the German TIGER corpus (Seeker and Kuhn, 2012). Following Kingsbury and Palmer's (2002, p. 1991) approach applied in the making of PropBank, ten random instances of 20 of the most frequent verbs were used to identify tendencies of the interaction of syntactic and semantic characteristics of these verbs.[6] The list of verbs comprises *bleiben* 'stay', *bringen* 'bring', *finden* 'find', *fordern* 'demand', *führen* 'lead', *geben* 'give', *gehen* 'go', *gelten* 'apply', *halten* 'hold', *heißen* 'mean', *kommen* 'come', *lassen* 'let', *liegen* 'lie', *machen* 'make', *nehmen* 'take', *sehen* 'see', *setzen* 'set', *stehen* 'stand', *stellen* 'put/provide', and *zeigen* 'show'. We applied two filters in the verb type and instance selection: We excluded two highly frequent verb types (*sagen* 'say/tell', *meinen* 'mean/think'), because they initiate an act of speech such that their argument structure often ranges over more than one sentence and is therefore not captured in the annotation of syntactic dependencies. In addition, we required the target verb to be the finite verb of the sentence, excluding all verbal complexes with auxiliary or modal verbs. The actual annotation process was performed with WebAnno (Eckart de Castilho et al., 2016) as shown in the screenshot in figure 1. For each verb, the target sentences were presented to the annotator together with its two preceding context sentences.

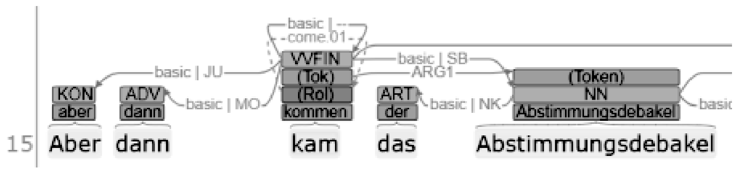

Figure 1: Annotation of *And then there was the voting debacle* in WebAnno.

4.2 Annotation

For each German verb we annotated the English sense from PropBank that matches semantic content and argument structure best, e. g. *come.01* for *kam* in figure 1. Considering the argument structure of the chosen sense, we then identified and assigned the semantic roles to the arguments realized in the sentence according to PropBank's annotation guidelines.

ARG0:	agent
ARG1:	patient
ARG2:	instrument, benefactive, attribute
ARG3:	starting point, benefactive, attribute
ARG4:	ending point
ARGM-PRR:	true predicate
ARGM-PRD:	secondary predication

Table 1: Tagset *semantic roles* following the PropBank annotation guidelines (Palmer et al., 2015, p. 3).

In addition to the core arguments, we included the roles ARGM-PRR and ARGM-PRD to label the true predicate in light verb constructions as well as cases of secondary predications. The annotations link the predicate to the argument heads congruent with the syntactic dependency relations. In the example in figure 1 the verb is *kommen* 'to come' in an abstract (temporal) meaning of motion, which realizes only one of the four argument roles according to PropBank's general entry (see figure 2).[7]

[6] Ten instances are too few to capture the full polysemy of highly frequent verbs. One reviewer pointed out that many of the verbs on our list have many more than 20 or 30 frames in SALSA. Burchardt et al. (2006) argue for a sample size of 20 as a "reasonable compromise between keeping the effort practicable and encountering the most important senses."

[7] `http://verbs.colorado.edu/propbank/framesets-english-aliases/come.html`.

Arg1-PPT: *entity in motion / 'comer'* (vnrole: 51.1-theme)
Arg2-EXT: *extent -- rare*
Arg3-DIR: *start point*
Arg4-GOL: *end point*

Figure 2: PropBank roleset of the verb sense *come.01: motion*.

The instance of *kommen* ‘come’ in Figure 1 has two syntactic dependencies: the subject (SB), *das Abstimmungsdebakel* ‘the voting debacle’ and a modifier (MO), *dann* ‘then’. Only the temporal entity in motion (ARG1) is realized in the sentence, which is structurally congruent with the subject (SB). There is no semantic role that aligns with the adverbial modifier (MO), which is not annotated on the semantic level.

The annotation was performed by two native speakers of German, both students of linguistics and proficient L2 speakers of English.

4.3 Results

Inter-annotator agreement The overall inter-annotator agreement between the two annotators on roleset assignment was 67 % (percent agreement, macro-averaged over 20 verbs). It turned out that in the cases of *machen* ‘make’ and *setzen* literal ‘sit’ both annotators decided on different main rolesets: make.01 “create” vs. make.02 “cause (to be)” and rely.01 “need” vs. count.01 “enumerate” respectively.[8] Ignoring these two verbs, the average percent agreement is 83 %. Closer inspection showed that even if the annotators decided on different rolesets, they often agreed in the argument assignment: We observe 84 % agreement over all 20 verbs; without *machen* and *setzen* agreement increases to 86 %.

Distributional results In the following we took the annotations of one of the annotators as reference data. In total, she assigned 380 annotations to 199 sentences (1.91 arguments per verb on average). The corresponding rolesets in PropBank average 2.76 arguments per verb. This means that in the corpus 69% of the rolesets’ arguments were actually realized.

	SB	OA	OC	MO	PD	CVC	DA	OP	Sum
ARG0	109 (100)	0	0	0	0	0	0	0	109
ARG1	52 (30)	85 (48)	25 (14)	†9 (5)	0	*1 (1)	0	4 (2)	176
ARG2	0	0	3 (5)	48 (86)	1 (2)	0	4 (7)	0	59
ARG3	0	0	2 (9)	15 (68)	5 (23)	0	0	0	22
ARG4	0	0	0	7 (10)	0	0	0	0	7
ARGM-PRR	0	0	0	0	0	10 (10)	0	0	10
Sum	161 (42)	80 (23)	30 (8)	79 (21)	6 (2)	11 (3)	4 (1)	4 (1)	380

Table 2: Distribution of semantic roles according to TIGER dependencies (absolute frequencies, percentages in parentheses), †) could be reanalyzed as objects, *) syntax error.

Analyzing the applicability of using the English PropBank as a linguistic resource for the manual annotation of German, we observed that in 86 % of the annotated German verb instances, she found a matching set of roles as a direct translation in the PropBank database. Even though in 54 of the 380 annotations it was necessary to use a roleset that was not a direct translation of the German verb, a matching set was found for all sentences (e.g., write.01 for *stehen* ‘to stand’ with the only appearing argument ARG1,“the thing written”). For every verb she annotated on average 2.8 different sets of semantic arguments (±1.03), meaning that the average verb appeared in 2.8 different contexts.[9]

[8] In addition, there were three instances in these verb sets, in which the target verb occurred twice and there was a misunderstanding between the annotators over which particular instance to annotate.

[9] This number corresponds to the observation reported in Burchardt et al. (2006) with respect to the original FrameNet annotation of SALSA.

The two core arguments ARG0 and ARG1 (agent and patient) mostly appeared with subjects and objects, see table 2. Examining the exceptions to this more closely showed that the realizations as modifier (MO) are an artifact of the TIGER annotation guidelines. According to the guidelines, prepositional objects are only labeled OP when the verb cannot also initiate a modifier with the same preposition (Albert et al., 2003, p. 124). This is a very restrictive interpretation of the prepositional object function. The single realization of an ARG1 as a CVC (collocational verb construction) turned out to be due to an erroneous dependency structure.

Taking a closer look at the MO dependencies, we identified twelve out of the 79 cases that could be reanalyzed as prepositional objects (including all nine cases of ARG1).[10] For the remaining 67 cases, the phrases were confirmed as adverbials. According to valency theory, obligatory adverbials are part of the roleset of a verb, which would be an explanation for the high frequency of modifiers related to our semantic annotations. We did not try to verify this assumption because obligatory and optional adverbials are not distinguished in the TIGER dependencies.

5 Discussion

The findings show that the core roles ARG0 and ARG1 are mostly assigned to syntactic subjects and objects and that 68 % of the other roles' instances are aligned with modifiers. In the following we discuss five phenomena in which syntactic annotation did not fully support the semantic role assignment.

Prepositional objects As pointed out in section 4.3, the distinction between prepositional objects and modifiers is notoriously difficult even for human annotators. The TIGER guidelines solve this problem by means of a very rigid rule and a list of eligible verbs and their prepositions for OP dependencies. Cases of polysemous verbs in which the very same preposition can introduce a prepositional object or an adverbial depending on the verb's reading are not distinguished in TIGER. Both dependency relations are marked as MO, such as in the examples in (1) and (2).[11]

(1) Sie setzten auf$_{\mathrm{MO}}$ das Eigeninteresse der Union.
they counted on the self-interest of the Union
'They were counting on the Union's self-interest.' (prepositional object)

(2) sein Bild auf$_{\mathrm{MO}}$ die Frontseiten setzen
his picture on the front_pages put
'put his picture on the front pages' (adverbial)

Distributed arguments in coordination There is a discrepancy of 189 sentences and only 161 semantically annotated subjects in table 2. This is partly due to cases of coordination:[12] While PropBank's syntactic database, the Penn TreeBank, inserts traces for distributed arguments in coordinations, TIGER dependencies do not capture this information, as seen in example (3).[13]

(3) Einige flirten [...] mit Fremden und machen$_{\mathrm{PRED}}$ aus Neugier$_{\mathrm{MO.ARG2}}$ [...] keinen Hehl$_{\mathrm{OA.ARG1}}$.
Some flirt [...] with strangers and make of curiosity no secret
'Some flirt with strangers and make no secret of curiosity.'

In (3), the subject *Einige* 'Some' does not have a direct dependency link to the verb *machen* 'make'.

[10]For our definition of prepositional object, see Dürscheid (2012, p. 40).

[11]Note that Universal Dependencies (2.0) avoid the distinction of propositional objects and adverbials altogether by assigning uniformly an obl relation to the nominal head.

[12]Another reason for this discrepancy is the fact that expletive subject *es* 'it' is syntactically marked as EP (expletive) in TIGER and not as SB. In addition to 14 expletive subjects, we did not annotate the subjects in cases of light verb constructions, see below. One reviewer pointed out correctly that the latter should have been covered in our annotation.

[13]Our claim that TIGER does not include the information on distributed arguments in coordinations does not hold for the original constituent annotation of TIGER as one reviewer pointed out. However, we cannot rely on this information because it is not replicated by dependency parsers trained on TIGER.

Light verb constructions Another finding concerned the interaction of syntax and semantics in sentences where the verb does not carry the main predication meaning, specifically in light verb constructions.In those sentences, the predicative noun phrase or prepositional phrase which carries the main meaning is linked to the verbal head by the relation CVC (collocational verb construction) in the TIGER corpus. As Hwang et al. (2010) explain, a multilevel annotation including the arguments of the light verb and the true predicate is necessary for a complete annotation of these cases. Since the dependency annotations only scaffold the link of the arguments to the verbal head, i.e. the light verb, and not to the true predicate, there is a mismatch of dependency structure and semantic relations.

(4) Sie$_{\text{SB}}$ halten$_{\text{PRED}}$ ihre Alten$_{\text{OA}}$ in Ehren$_{\text{CVC.ARGM-PRR}}$.
They hold their elders in honour
'They cherish their elders.'

In (4), the German light verb construction in *Ehren halten* 'to cherish' contains a light verb *halten* 'to hold' and the CVC *in Ehren* 'in honor' which carries the main meaning of the predication.

Secondary predication In secondary predication there is an argument that "modifies another argument of the verb (describing its state during or after the event) more than it modifies the verb or event itself" (Palmer et al., 2015, p. 15), cf. Example (5).

(5) ... lässt$_{\text{PRED: LEAVE.12}}$ ihn$_{\text{OA.ARG1}}$ das böse Wort$_{\text{SB.ARG0}}$ von der Wiedergutmachungs-Industrie
... leave him the bad word from the reparation-industry
nicht kalt$_{\text{MO.ARG2-PRD}}$.
not cold
'... the bad word from the reparation industry is affecting him.'

In (5), the secondary predication *kalt* 'cold' describes the state in which the object *ihn* 'him' is after the event is over. The syntactic annotation relates the modifying *kalt* to the main verb and not to the object. On the semantic level *kalt* should also be linked to the object.

Inherently reflexive verbs In German, some verbs are compulsorily accompanied by a semantically empty pronoun in reflexive form (it morphologically agrees in person with the subject) as *in sich nehmen* 'to take'. English equivalents do not have these dummy pronouns, hence there are no argument roles for them in the PropBank sets. In TIGER, they are annotated as accusative objects. This leads to a further mismatch in syntactic dependencies and semantic relations.

6 Conclusion

We performed an evaluation study on the semantic annotation of a set of 20 verb types, by manually annotating semantic PropBank senses and argument roles on top of TIGER dependency annotations in 199 German sentences. Our inter-annotator comparison showed that the use of the English PropBank as a knowledge base for manual annotation of German was feasible but that our annotators, who have a good L2 competence of English, would have needed more guidance for inter-subjectively more reliable decisions on specific English verb senses. Our annotation study showed that in a controlled setting syntactic dependencies can provide a useful scaffolding for the annotation of semantic roles. We also identified phenomena in which syntactic dependencies systematically do not support the actual semantic relations. It will be necessary to analyze more examples and different types of verbs to further our understanding.

Acknowledgments

We would like to thank the anonymous reviewers for their very detailed and helpful comments. We would also like to thank Lea Röseler for her annotations, and Piklu Gupta for improving our English.

List of abbreviations

SB subject
OA accusative object
OC clausal object
MO modifier
PD predicate
CVC collocational verb construction
DA dative (object and 'free' dative)
OP prepositional object
PRR predicating relation
PRD secondary predication

References

Albert, S., J. Anderssen, R. Bader, S. Becker, T. Bracht, S. Brants, T. Brants, V. Demberg, S. Dipper, P. Eisenberg, S. Hansen, H. Hirschmann, J. Janitzek, C. Kirstein, R. Langner, L. Michelbacher, O. Plaehn, C. Preis, M. Puel, M. Rower, B. Schrader, A. Schwartz, G. Smith, and H. Uszkoreit (2003, July). TIGER Annotationsschema.

Baker, C. F., C. J. Fillmore, and J. B. Lowe (1998). The Berkeley FrameNet Project. In *Proceedings of the 17th International Conference on Computational Linguistics*, Volume 1 of *COLING '98*, Stroudsburg, PA, USA, pp. 86–90.

Beißwenger, M. and A. Storrer (2008). 21. corpora of computer-mediated communication. *Corpus Linguistics. An International Handbook. Series: Handbücher zur Sprach-und Kommunikationswissenschaft/Handbooks of Linguistics and Communication Science. Mouton de Gruyter, Berlin*.

Brants, S., S. Dipper, P. Eisenberg, S. Hansen-Schirra, E. Knig, W. Lezius, C. Rohrer, G. Smith, and H. Uszkoreit (2004). TIGER: Linguistic interpretation of a German corpus. *Research on language and computation 2*(4), 597–620.

Burchardt, A., K. Erk, A. Frank, A. Kowalski, S. Padó, and M. Pinkal (2006). The SALSA corpus: a German corpus resource for lexical semantics. In *Proceedings of LREC*, pp. 969–974.

Dowty, D. (1991). Thematic Proto-Roles and Argument Selection. In: Language. *Language 67*(3), 547–619.

Dürscheid, C. (2012). *Syntax: Grundlagen und Theorien* (6 ed.). Vandenhoeck & Ruprecht.

Eckart de Castilho, R., É. Mújdricza-Maydt, S. M. Yimam, S. Hartmann, I. Gurevych, A. Frank, and C. Biemann (2016). A Web-based Tool for the Integrated Annotation of Semantic and Syntactic Structures. In *Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH)*, Osaka, Japan, pp. 76–84.

Fillmore, C. J. (1968). The Case for Case. In E. Bach and R. T. Harms (Eds.), *Universals in Linguistic Theory*, pp. 1–88. New York: Holt, Rinehart, and Winston.

Fillmore, C. J. (1977). The case for case reopened. *Syntax and semantics 8*(1977), 59–82.

Gildea, D. and D. Jurafsky (2002). Automatic labeling of semantic roles. *Computational linguistics 28*(3), 245–288.

Hajic, J., M. A. Mart, L. Marquez, J. Nivre, J. tpnek, S. Pad, and P. Strak (2012). 2009 CoNLL Shared Task Part 1 LDC2012t03. Web Download. Technical report, Linguistic Data Consortium, Philadelphia.

Hajič, J., M. Ciaramita, R. Johansson, D. Kawahara, M. A. Martí, L. Màrquez, A. Meyers, J. Nivre, S. Padó, J. Štěpánek, P. Straňák, M. Surdeanu, N. Xue, and Y. Zhang (2009, June). The conll-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task*, Boulder, Colorado, pp. 1–18.

Hamp, B. and H. Feldweg (1997). Germanet-a lexical-semantic net for german. *Automatic information extraction and building of lexical semantic resources for NLP applications*.

Hartmann, S., É. Mújdricza-Maydt, I. Kuznetsov, I. Gurevych, and A. Frank (2017). Assessing srl frameworks with automatic training data expansion. In *Proceedings of the 11th Linguistic Annotation Workshop*, pp. 115–121.

Hwang, J. D., A. Bhatia, C. Bonial, A. Mansouri, A. Vaidya, N. Xue, and M. Palmer (2010). Propbank annotation of multilingual light verb constructions. In *Proceedings of the Fourth Linguistic Annotation Workshop*, LAW IV '10, Stroudsburg, PA, USA, pp. 82–90. Association for Computational Linguistics.

Kingsbury, P. and M. Palmer (2002). From treebank to propbank. In *Proceedings of LREC*, pp. 1989–1993.

Kipper Schuler, K. (2005). *VerbNet: A broad-coverage, comprehensive verb lexicon*. Ph. D. thesis, University of Pennsylvania.

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *MT summit*, Volume 5, pp. 79–86.

Levin, B. (1993). *English Verb Classes and Alternations*. Chicago: The University of Chicago Press.

Mújdricza-Maydt, É., S. Hartmann, I. Gurevych, and A. Frank (2016, Mai). Combining semantic annotation of word sense & semantic roles: A novel annotation scheme for verbnet roles on german language data. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pp. 3031–3038.

Padó, S. and M. Lapata (2005). Cross-linguistic projection of role-semantic information. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pp. 859–866. Association for Computational Linguistics.

Palmer, M., C. Bonial, J. Hwang, J. Bonn, K. Conger, and O. Babko-Malaya (2015). English propbank annotation guidelines.

Palmer, M., D. Gildea, and P. Kingsbury (2005). The Proposition Bank: A Corpus Annotated with Semantic Roles. *Computational Linguistics Journal 31*(1).

Seeker, W. and J. Kuhn (2012). Making Ellipses Explicit in Dependency Conversion for a German Treebank. In *Proceedings of LREC*, Istanbul, Turkey, pp. 3132–3139.

The Semantic Annotation of Measurable Quantitative Information

Tianyong Hao
South China Normal University
haoty@m.scnu.edu.cn

Shanshan Liu
Guangdong University of Foreign Studies, China
shan63333@hotmail.com

Haitao Wang*
China National Institute of Standardization
wanght@cnis.gov.cn

Xinyu Cao
China National Institute of Standardization
caoxy@cnis.gov.cn

Kiyong Lee
Korea University
ikiyong@gmail.com

Abstract

Measurable quantitative information (MQI), covering the pivotal measurable attributes associated with various types of entities, is omnipresent in scientific-technological text. Aiming to represent quantitative information in a normative and concise way, this paper analyzes the characteristics of MQI and the features of quantity in general, and proposes a new version of a markup language for the annotation of measurable quantitative information (QML), compatible with ISO 24617 semantic annotation standards. To visualize the annotation framework, an online annotation system was developed that is presented here.

1 Introduction

Along with the fast-developing applications of electronic systems, in recent years, the amount of text data has increased exponentially in various branches of industry (Berkout et al., 2018; Bigeard et al., 2018). Measurable quantitative information (MQI), which concerns the magnitude aspect of quantity, is ubiquitous in textual data (Hao et al., 2018). For instance, industry research reports that contain market scales, competition patterns and other market characteristics have a great deal of quantitative information targeting the economic domain. In view of the importance of quantitative information in data analysis and tendency forecasting, the reliable extraction and representation of MQI from unstructured data have received considerable attention (Hao et al., 2017; Madaan et al., 2016).

High-technology industries and academic communities have developed a range of applications of Natural Language Processing (NLP) and Information Extraction (IE) for extracting and representing MQI from free text (Murata et al., 2008; Wang et al., 2017). However, owing to the absence of a standardized representation format for annotating MQI, most of these systems are not interchangeable or interoperable. This issue continues to be a challenge.

The construction of an annotation scheme for MQI is in progress in ISO/TC 37/SC 4/WG 2 (Semantic Annotation). In this paper, recent advances in defining a markup language for annotating measurable quantitative information (QML), which facilitates the processing of MQI in scientific and technical language, are discussed and illustrated in some detail. QML as a specification language for the annotation of MQI has two levels of syntactic definition: the level of abstract syntax QML*as* and the level of concrete syntax QML*csx*. Based on an abstract syntax, a particular concrete syntax defines a representation format for the annotation structures defined by the abstract syntax (and associated semantics).

As Part 11 of ISO 24617 Language resource management - Semantic Annotation Framework (SemAF), QML is consistent with the general principles of the Linguistic Annotation Framework (LAF) (ISO, 2016), and with the principles for semantic annotation laid down in ISO 24617-6 (ISO, 2016); moreover it is compatible with ISO standards that are relevant to the annotation of numerals and units, such as ISO-TimeML (ISO, 2012b) and ISO-Space (ISO, 2014a). The XML-based concrete syntax of QML (QML*csx*) in particular adopts XML-based TEI serialization. This paper reports recent progress in the specification of QML.

2 Measurable Quantitative Information

2.1 Quantity and unit

There are two aspects of quantity, multitude (e.g., a mass of people and a battalion of ants) and magnitude (e.g., size and scale). In this paper, different from the definition of quantity in Cartwright (1970), quantity is regarded as a magnitude property, which is consistent with ISO 80000-1:2009 Quantities and Units (ISO, 2009). The magnitude aspect of quantity (*Q*) is expressed by a pair <*n*, *u*> consisting of a numeral (*n*) and a unit of measurement (*u*). Different kinds of quantities are associated with different units by convention. According to ISO 80000-1:2009 Quantities and Units (ISO, 2009), quantities can be divided into two categories, base quantities and derived quantities. The main distinction between them is that a base quantity can be measured in terms of a basic unit, whereas a derived quantity requires a complex unit for being measured. For example, a distance is a base quantity (measured with unit of length), whereas a speed is a derived quantity (measured with a complex unit of length). Quantities in this paper have at least one dimension to show the dependence of a quantity on the base quantities of a system of quantities referred to in ISO standard 80000 - 1:2009 Quantities and Units (ISO, 2009), and it can be expressed by equation (1). In the equation, L, M, T, I, Θ, N and J are seven basic dimensions defined by the International System of Units (SI) with $\partial, \beta, \gamma, \delta, \varepsilon, \xi$ and η as exponents, respectively.

$$Q == L^{\partial} M^{\beta} T^{\gamma} I^{\delta} \Theta^{\varepsilon} N^{\xi} J^{\eta} \tag{1}$$

As defined in ISO 80000-1:2009, the term "unit" is used for the specification of measurement of physical quantities, which is defined and adopted by convention. For example, "meter", "gram", "litre", and "μmol/kg" are units by definition, while "bottle", "apple", and "coffee", as in "two bottles of milk", "a box of apples", and "two coffees" are not regarded as units unless they are accurately defined by convention or agreement.

Units can be divided into three categories: base units, derived units and compound units. Referring to the SI brochure (2001) published by the General Conference on Weights and Measures (CGPM), there are seven base units associated with seven base quantities.

Table 1. The associations and differences among base units and derived units.

Base units		**Derived units**	
Base units (unit symbols)	Associated quantity names (dimension symbol)	Singular derived units (unit symbols)	Compound derived units (unit symbols)
meter (m)	Length (L)	kilometre (km)	kilometre per minute (km/min)
kilogram (kg)	Mass (M)	gram (g)	gram per stere (gram/m^3)
second (s)	Time (T)	hour (h)	kilowatt per hour (kw/h)
ampere (A)	electric current (I)	microampere (μA)	ampere per square metre ($A \cdot m^{-2}$)
Kelvin (K)	thermodynamic temperature (Θ)	degree Celsius (℃)	temperature gradient, kelvin per metre ($m^{-1} \cdot K$)
Mole (mol)	amount of substance (N)	micromole (μmol)	mole per cubic metre ($mol \cdot m^{-3}$)
Candela (cd)	luminous intensity (J)	lux (lx)	lumen per square metre (lm/m^2), candela/foot² (cd/ft^2)

Derived units are the measurement units for derived quantities, and they have two sub-categories, singular derived units and compound derived units, as defined in ISO 80000-1:2009 (ISO, 2009) and ISO/WD 24617-11 SemAF-Part 11 MQI (ISO, 2019). A singular derived unit is a single measurement unit for a derived quantity while a compound derived unit is a combination of different measurement units for a derived quantity. The associations and differences among them are illustrated in Table 1.

2.2 The features of MQI

In the measurable quantitative information markup language (QML), each unit, either base or derived, is indivisible and is treated as an unanalyzed concept. Derived units are not disassembled into their components to show their internal structures. According to ISO/WD 24617-11(ISO, 2019), this is based on the following arguments: 1) Some frequently used compound derived units, e.g., “km/h” (LT^{-1}), are easily understood for ordinary uses. 2) Certain special domain units, e.g., “mL/min/((173/100).m^2)” for Estimated Glomerular Filtration Rate (eGFR) in the medical domain, contain non-base units thus they cannot be disassembled during conversion. 3) There are incomplete units with missing parts during practical text processing, e.g., “mg/kg/” is an incomplete unit in drug usage with the part “d” missing.

MQI focuses on the annotation of the measurable attributes of entities. For example, “HbA1c from 5 to 8%”. MQI also provides a strategy to annotate comparisons of measures, rather than counting objects only (Rothstein, 2017). For example, “HbA1c not less than 5.8%”. In addition, MQI covers certain types of complex cases of numerals and units. For example, “glycosylated hemoglobin > 1.2 times the upper limit of normal”. MQI is expected to facilitate the identification of normalized numerals and units, as measurable attributes of associated entities.

3 Specification of QML

3.1 Metamodel

As an ISO semantic annotation language, QML has a three-part definition including an abstract syntax specifying conceptual annotation structures (QML*as*) and a set of equivalent concrete syntaxes (QML*cs*). Based on our previous work as described in Hao et al. (2018), the metamodel of QML is slightly simplified and improved. The elements “communicative segments” and “quantitative relation” in the previous version of the model are replaced by “source data” and “comparison” respectively, and the element “dimension” is eliminated. The new metamodel is as shown in Figure 1.

The new metamodel contains six types of elements: source data, markable, entity, measure, comparison, and link. Source data are input to the annotation of MQI, and markable expressions are extracted from source data. Entity, measure, and comparison are three types of basic elements, while two types of link are distinguished: measure link and comparison link.

Compared to the previous version, the metamodel, is more concise and clear. Three types of basic elements cover basic aspects of MQI: 1) the element “entity” is any object which has a measurable quantity as one of its properties; 2) the element “measure” represents a measurable quantity of an entity in terms of three attributes: @numeral, @unit, and @type; 3) the element “comparison” which is associated with markables such as “greater than”, “<=”, or “at least” has a functional status of comparing two or more measures. The two types of link describe the kinds of linkages amongst measures and entities, and the comparison relations amongst measures, respectively: 1) a “measure link” relates a measure to the quantitative property of an entity; 2) a “comparison link” relates one measure to another.

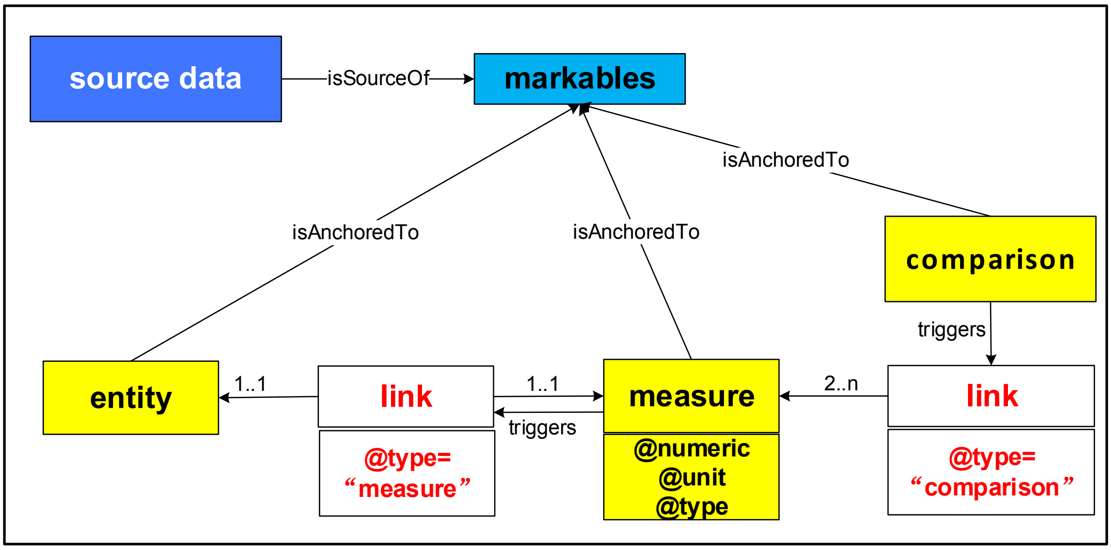

Figure 1 New metamodel of QML for SemAF-Part 11 MQI.

3.2 Abstract syntax

As presented in the ISO/WD 24617-11 document (SemAF-Part 11, MQI, ISO, 2019), the abstract syntax QML*as* of the semantic annotation of MQI is defined as a triple <*B*, *R*, @>. *B* denotes a non-empty set of basic element types: entity, measure, and comparison. *R* denotes a non-empty set of link types: measure link and comparison link. @ is an assignment of attributes and their values to each of the basic element types in *B* and each of the link types in *R*. @ (element) is a list of required attributes of an element, e.g., @ (measure) is <@num, @unit, @type>. @ (link) has three parts containing an entity structure, a set of entity structures and associated relations, e.g., @ (link) is <@measureID, @appliedTo, @relType>.

To illustrate the abstract syntax QML*as*, consider the example *"White blood cell count > 14.0 X 109 / L."*. In this sentence, "*White blood cell*" describes an entity. "*14.0 X 109 / L*" describes a measure consisting of two attributes @numeral ("*14.0 X 109*") and @unit ("*/ L*"). ">" describes a comparison relation ("larger than"). A measure link and a comparison link are triggered by the word "*count*" and by ">", respectively.

3.3 Concrete syntax

Alternative semantically equivalent concrete syntaxes of QML (QML*cs*) are allowed based on the same abstract syntax QML*as* following ISO 24617-6, similar to the strategy proposed in (Bunt, 2010). In this paper, a concrete syntax is proposed named QML*csx*, which adopts the extensible markup language (XML) as its representation language. The specification of the concrete syntax QML*csx* is as follows:

1) Tags

Each basic element and link relation is represented in a serialized XML format. Each of them is uniquely identified by an ID prefix followed by an integer. A list of XML elements represents an annotation structure as a single XML structure. Table 2 provides a description of each tag.

Table 2. Tags defined in QML*csx*.

Tags	Explanations
Root tag	
<MQI>	XML root tag
Basic element tags	
<entity>	entity to which a measure applies
<measure>	unitized numeral quantities
<comparison>	triggers a link comparing measures
Link tags	
<mLink>	the tag of measure link that relates a measure to an entity and is triggered by a measure
<cLink>	the tag of comparison link that relates a measure to another or other more measures

2) Attributes of root tag and basic element tags

The specific attributes of each tag are shown in Table 3.

Table 3. The data formats of attributes defined in QML*csx*.

	<MQI>	**<entity>**	**<comparison>**	**<measure>**
@identifier	mqi + PI *	x + PI *	cp + PI *	me + PI *
	Note: PI denotes positive integer.			
@target	IDREF| CDATA *	IDREF| CDATA *	IDREF| CDATA *	IDREF| CDATA *
	Note: @target of <MQI> refers to the ID of a sequence of source data or such a segment sequence directly quoted, while others refer to the ID of a sequence of source data or a markable.			
@lang	CDATA	*Not applicable*	*Not applicable*	*Not applicable*
	Note: @lang refers to ISO 639 standard on language codes.			
@medium	CDATA	*Not applicable*	*Not applicable*	*Not applicable*
	Note: text, video, image, etc.			
@source	CDATA	*Not applicable*	*Not applicable*	*Not applicable*
@type	*Not applicable*	CDATA *	CDATA *	CDATA *
	Note: @type of different tags have distinct kinds of values, such as @type of <entity> describes ontological types like "sportFacilite", and "medicalConcept"; @type of <measure> denotes the dimension of quantity, such as L-dimensional types: length, width, height, etc.; @type of <comparison> denotes a mathematical operator, e.g., "lower than" and "equal to".			
@comment	CDATA	CDATA	CDATA	CDATA
@num	*Not applicable*	*Not applicable*	*Not applicable*	real numbers *
@unit	*Not applicable*	*Not applicable*	*Not applicable*	CDATA
	Note: @unit is not specified when a measure is associated with the degree of efficiency such as John's score is 95 or some grade range from 0 to 1.			

Note: Attributes with "" are required. Otherwise, they are optional.*

3) Attributes of relation links

There are two types of relation links, tagged <mLink> and <cLink>, in the XML-based

QML*csx*. According to ISO 24617-6 SemAF Principles, the structure of links can be represented by a triple <η, E, ρ>, where η is an entity structure, E is a set of entity structures, and ρ is a relation from η to the members of E. An entity structure consists of a markable and the corresponding annotated semantic information. For measure link <mLink> in QML*csx*, the entity structure η is a measure, corresponding to the value of an @measureID attribute; the set of entity structures E corresponds to the value of an @appliedTo attribute, and the relation ρ to that of a @relType attribute. In a comparison link <cLink> both the entity structure η and the members of the set of entity structures E are measures (corresponding to the values of @measure1 and @measure2, respectively) and the relation ρ corresponds to the value of the @relType attribute.. The attributes of <mLink> and <cLink> are specified in Table 4 and Table 5, respectively.

Table 4. The attributes of <mLink>.

Attributes	**Data types**	**Values**
@identifier	ID*	meL + PI
@measureID	IDREF*	ID of <measure>
@ appliedTo	IDREFS*	IDS of <entity>s to which the value of <measure> applies
@ relType	“measures”*	follows the specification of ISO 24617-1:2012 ISO-TimeML
@comment	CDATA	*Not applicable*

Note: PI in the table denotes positive integer. Attributes with “” are required. Otherwise, they are optional.*

Table 5. The attributes of <cLink>.

Attributes	**Data types**	**Values**
@identifier	ID*	coL + PI
@ measure1	IDREF*	ID of <measure>
@ measure2	IDREFS*	ID of < measure>
@ relType	CDATA*	type of a relation that compares one measure to another measure or other measures
@comment	CDATA	*Not applicable*

Note: PI in the table denotes positive integer. Attributes with “” are required. Otherwise, they are optional.*

4) Illustration of QML*csx*

Each sentence that contains measurable quantitative information is pre-processed through word segmentation. In order to assign each anchored word token as value of the attribute @target, we transformed words in sentences into a list of target words with a unique ID such as *w1*, *w2*, etc.

For example, the word segmentation of the sentence, “*White blood cell count > 14.0 X 109 / L*”, is represented as below.

```
<wordSeg xml:id="ws1" target="#1a" lang="en">
    White_w1 blood_w2 cell_w3 count_w4 >_w5 14.0_w6 X_w7 109_w8 /_w9
    L_w10 ._p1
</wordSeg>
```

Here are two examples of word segmentation and annotation of MQI, as represented in QML*csx*.

Example 1: (Measure Link)

a. <text xml:id = “*t1*” > fasting blood sugar level is 125 mg/dl. </text>

b. <wordSeg xml:id=“*ws1*” target=“#t1” lang=“en” >

fasting_w1 blood_w2 sugar_w3 level_w4 is_w5 125_w6 mg/dl_w7 ._p1

</wordSeg>

c. <MQI xml:id=“*qi1*” target=“#ws1” >

<entity xml:id=“*x1*" target=“#w1, #w2, #w3" type=“*medicalConcept*" />

<measure xml:id=“*me1*" target=“*#w4, #w6, #w7*” num=“*125*" unit=“*mg/dl*" type=“*level*” />

<mLink xml:id=“*meL1*" measureID =“*#me1*” appliedTo =“*#x1*" relType=“*measures*" />

</MQI>

Example 2: (Comparison Link)

a. <text xml:id = “*t2*” > White blood cell count > 14.0 X 109 / L. </text>

b. <wordSeg xml:id=“*ws2*” target=“#t2” lang=“en” >

White_w1 blood_w2 cell_w3 count_w4 >_w5 14.0_w6 X_w7 109_w8 /_w9 L_w10 ._p1

</wordSeg>

c. <MQI xml:id=“*qi2*” target=“#ws2” >

<entity xml:id=“*x1*" target=“#w1, #w2, #w3" type=“*medicalConcept*" />

<measure xml:id=“*me1*" target=“ ” num=“ ” unit=“ ” type=“ ” />

<measure xml:id=“*me2*" target=“*#w4, #w6, #w7, #w8, #w9, #w10*” num=“*14.0 X 109*" unit=“*/L*" type=“*count*” />

<comparison xml:id=“*cp1*” target =“*#w5*” type=“*greaterThan*” >

<cLink xml:id=“*coL1*" measure1=“*#me1*” measure2=“*#me2*" relType=“*greaterThan*” />

<mLink xml:id=“*meL1*" measureID=“*#me1*” appliedTo=“*#x1*" relType=“*measures*" />

</MQI>

4 Visualization of QML*csx*

In order to visualize QMLcsx annotation representations, an online annotation system was designed with the development environment of Python 3.6 and Django 2.1.4. Currently, the system provides the representation of word segments and the annotation of measurable quantitative information using QMLcsx in English and Chinese. Other languages will be supported in the future. The system is publically accessible at ‘http://47.102.207.52:83/semantic_tag/’.

During annotation with the online system, a user may enter a text containing measurable quantitative information. By clicking relevant buttons, each basic element defined in QML*csx* is automatically annotated with colors. For example, a basic element <entity> is annotated with a reddish color by clicking the button “Entity”. After that, the user may select types of entity, types of comparison and types of measure. Eventually, the system outputs annotation results in XML using QML*csx*. A screenshot of the system for the example (‘*White blood cell count > 14.0 X 109 / L.*’) is shown in Figure 2. Please note that the annotations in the system can be adjusted to adapt to the concrete syntax QML*csx*.

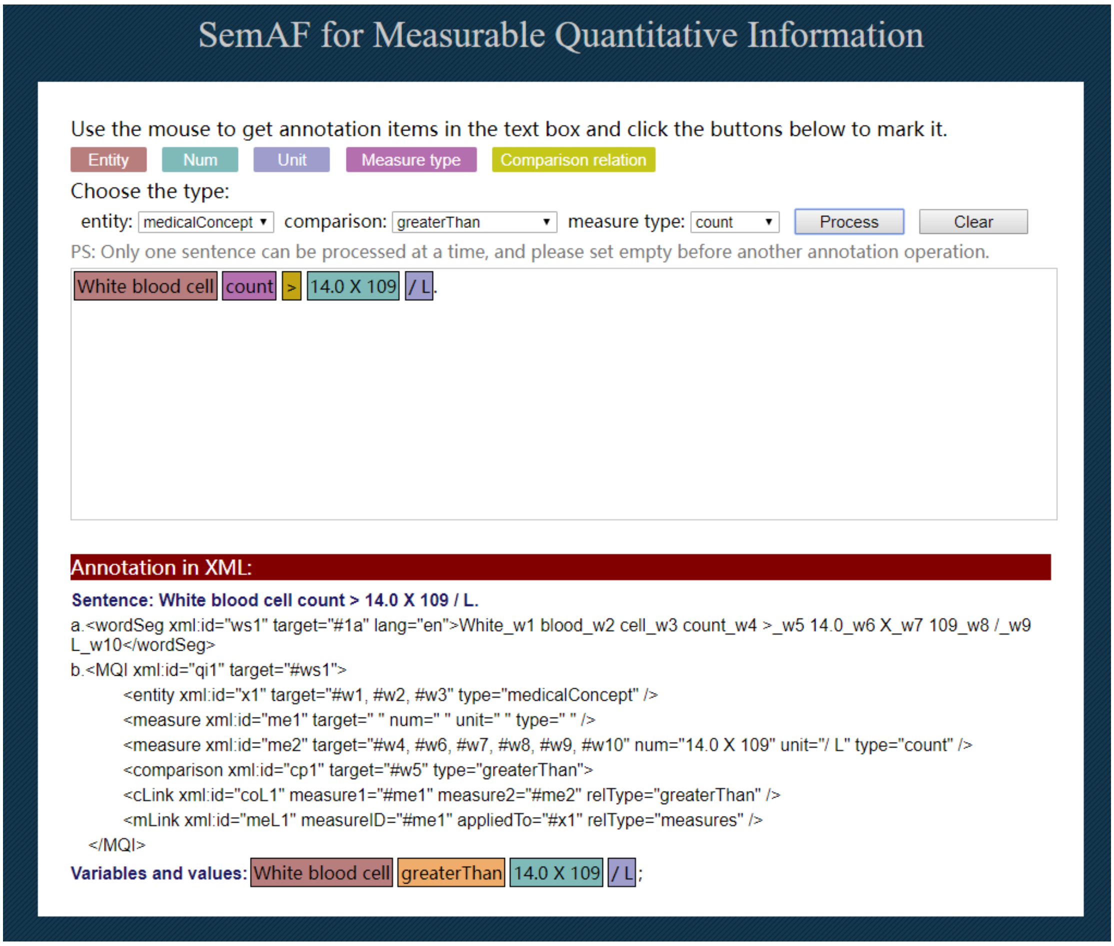

Figure 2 Screenshot of online annotation system using QML*csx*.

5 Conclusion

Quantity as the crucial component of measurable quantitative information (MQI) presents a high degree of complexity in scientific and technical text. Focusing on the magnitude aspect of quantity, this paper reports recent work on the abstract syntax and concrete syntax of the specification language QML, aiming at the applicability of semantic annotation for MQI in various domains. In addition, the categories of quantities and the features of units have been analyzed to illustrate the characteristics of MQI. As an annotation scheme, QML is designed to be compatible with other ISO standard annotation frameworks and is still in progress to become part of ISO standard 24617, and to become widely adopted in industry.

6 Funding

This paper is supported by National Natural Science Foundation of China (No. 61772146).

References

Berkout, O.V., Cathey, A.J., Kellum, K.K. (2018). Scaling-Up Assessment from a Contextual Behavioral Science Perspective: Potential Uses of Technology for Analysis of Unstructured Text Data. *Journal of Contextual Behavioral Science,* June 2018. DOI: 10.1016/j.jcbs.2018.06.007.

Bigeard, E., Jouhet, V., Mougin, F., et al. (2017). Automatic Extraction of Numerical Values from Unstructured Data in EHRs. *Studies in Health Technology & Informatics*, 210:50-54.

Bunt, H. (2010). A Methodology for Designing Semantic Annotation Languages Exploiting Semantic-syntactic ISO-morphisms. In *Proceedings of ICGL 2010, the Second International Conference on Global Interoperability for Language Resources*, Hong Kong, edited by Alex Fang, Nancy Ide and Jonathan Webster, pp. 29-45.

Cartwright, H. M. (1970). Quantities. *The Philosophical Review* 75 (1), 25-42.

Hao, T., Wei, Y., Qiang, J., Wang, H., Lee, K. (2017). The Representation and Extraction of Quantitative Information. In *Proceedings of the 13th Joint ISO-ACL Workshop on Interoperable Semantic Annotation (ISA013)*, Montpellier, edited by Harry Bunt, pp.74-83.

Hao, T., Wang, H., Cao, X., et al. (2018). Annotating Measurable Quantitative Information in Language: for an ISO Standard. In Proceedings of the 14th Joint ISO-ACL Workshop on Interoperable Semantic Annotation (ISA-14), Santa Fé (New Mexico), edited by Harry Bunt, pp: 69-75.

Madaan, A., Mittal, A., Mausam, Ramakrishnan, G., and Sunita Sarawagi, S. (2016). Numerical Relation Extraction with Minimal Supervision. In *Proceedings 30th AAAI Conference on Artificial Intelligence*. AAAI Press. Available at: <https://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12486>.

ISO (2016). ISO 24617-6 Language Resource Management - Semantic Annotation Framework – Part 6: Principles of Semantic Annotation (SemAF Principles) International Organization for Standardization ISO, Geneva.

ISO (2012b). ISO 24617–1:2012(E) Language Resource Management – Semantic Annotation Framework (SemAF) – Part 1: Time and Events. International Organization for Standardization ISO, Geneva.

ISO (2014). ISO 24617–7:2014(E) Language Resource Management – Semantic Annotation Framework (SemAF) –Part 7: Spatial Information. International Organization for Standardization ISO, Geneva.

ISO (2009). ISO 80000:2009 (E) Quantities and Units - Part 1: General. International Organization for Standardization ISO, Geneva.

ISO (2019). ISO/WD 24617-11 Semantic Annotation Framework (SemAF) - Part 11: Measurable Quantitative Information (MQI). International Organization for Standardization ISO, Geneva.

Murata, M., Ma, Q., Torisawa, K., Iwatate, M., Shirado, T., Ichii, K. and Kanamaru, T. (2008). Extraction and Visualization of Numerical and Named Entity Information from a Large Number of Documents. In *Proceedings of the 4th International Conference on Natural Language Processing & Knowledge Engineering (NLPKE)*, pp. 1-8.

Rothstein, S. (2017). *Semantics for Counting and Measuring*. Cambridge University Press, Cambridge, UK.

Taylor, B. N., Thompson, A. (2001). *The International System of Units (SI).* US Department of Commerce, Technology Administration, National Institute of Standards and Technology, TEI P5: Guidelines for Electronic Text Encoding and Interchange. (2016). The TEI Consortium.

Wang, Y., Wang, L., Rastegar-Mojarad, M., Moon, S., Shen, F., Afzal, N., Liu, S., Zeng, Y., Mehrabi, S., Sohn, S., Liu, H. (2017). Clinical Information Extraction Applications: A Literature Review. *Journal of Biomedical* Informatics 77, 34-49.

XML Schema Part 2: Datatypes, Second Edition. (2004). W3C Recommendation.

Author Index