A Domain Agnostic Approach to Verbalizing n-ary Events without Parallel Corpora

Bikash Gyawali
Université de Lorraine/LORIA
Nancy, France
bikash.gyawali@loria.fr

Claire Gardent
CNRS/LORIA
Nancy, France
claire.gardent@loria.fr

Christophe Cerisara
CNRS/LORIA
Nancy, France
christophe.cerisara@loria.fr

Abstract

We present a method for automatically generating descriptions of biological events encoded in the KB BIO 101 Knowledge base. We evaluate our approach on a corpus of 336 event descriptions, provide a qualitative and quantitative analysis of the results obtained and discuss possible directions for further work.

1 Introduction

While earlier work on data-to-text generation heavily relied on handcrafted linguistic resources, more recent data-driven approaches have focused on learning a generation system from parallel corpora of data and text. Thus, (Angeli et al., 2010; Chen and Mooney, 2008; Wong and Mooney, 2007; Konstas and Lapata, 2012b; Konstas and Lapata, 2012a) trained and developed data-to-text generators on datasets from various domains including the air travel domain (Dahl et al., 1994), weather forecasts (Liang et al., 2009; Belz, 2008) and sportscasting (Chen and Mooney, 2008). In both cases, considerable time and expertise must be spent on developing the required linguistic resources. In the handcrafted, symbolic approach, appropriate grammars and lexicons must be specified while in the parallel corpus based learning approach, an aligned data-text corpus must be built for each new domain. Here, we explore an alternative approach using non-parallel corpora for surface realisation from knowledge bases that can be used for any knowledge base for which there exists large textual corpora.

A more specific, linguistic issue which has received relatively little attention is the unsupervised verbalisation of n-ary relations and the task of appropriately mapping KB roles to syntactic functions. In recent work on verbalising RDF triples, relations are restricted to binary relations (called "property" in the RDF language) and the issue is therefore intrinsically simpler. In symbolic approaches dealing with n-ary relations, the mapping between syntactic and semantic arguments is determined by the lexicon and must be manually specified. In data-driven approaches, the mapping is learned from the alignment between text and data and is restricted by cases seen in the training data. Instead, we learn a probabilistic model designed to select the most probable mapping. In this way, we provide a domain independent, fully automatic, means of verbalising n-ary relations.

The paper is structured as follows. In Section 2, we discuss related work. In Section 3, we present the method used to verbalise KB events and their participants. In Section 4, we evaluate our approach on a corpus of 336 test cases, provide a qualitative and quantitative analysis of the results obtained and discuss possible directions for further work. Section 5 concludes.

2 Related Work

There has been much research in recent years on developing natural language generation systems which support verbalisation from knowledge and data bases.

Many of the existing KB Verbalising tools rely on generating so-called Controlled Natural Languages (CNL) i.e., a language engineered to be read and written almost like a natural language but whose syntax and lexicon is restricted to prevent ambiguity. For instance, the OWL verbaliser integrated in the Protégé tool is a CNL based generation tool, (Kaljurand and Fuchs, 2007) which provides a verbalisation of every axiom present in the ontology under consideration. Similarly, (Wilcock, 2003) describes an ontology verbaliser using XML-based generation. Finally, recent work by the SWAT project[1] has focused on pro-

[1] http://crc.open.ac.uk/Projects/SWAT

Proceedings of the 15th European Workshop on Natural Language Generation (ENLG), pages 18–27,
Brighton, September 2015.

ducing descriptions of ontologies that are both coherent and efficient (Williams and Power, 2010). In these approaches, the mapping between relations and verbs is determined either manually or through string matching and KB relations are assumed to map to binary verbs.

More complex NLG systems have also been developed to generate text (rather than simple sentences) from knowledge bases. Thus, the MIAKT project (Bontcheva and Wilks., 2004) and the ONTOGENERATION project (Aguado et al., 1998) use symbolic NLG techniques to produce textual descriptions from some semantic information contained in a knowledge base. Both systems require some manual input (lexicons and domain schemas). More sophisticated NLG systems such as TAILOR (Paris, 1988), MIGRAINE (Mittal et al., 1994), and STOP (Reiter et al., 2003) offer tailored output based on user/patient models. While offering more flexibility and expressiveness, these systems are difficult to adapt by non-NLG experts because they require the user to understand the architecture of the NLG systems (Bontcheva and Wilks., 2004). Similarly, the NaturalOWL system (Galanis et al., 2009) has been proposed to generate fluent descriptions of museum exhibits from an OWL ontology. These approaches however rely on extensive manual annotation of the input data.

Related to the work discussed in this paper is the task of learning subcategorization information from textual corpora. Automatic methods for subcategorization frame acquisition have been proposed from general text corpora, e.g., (Briscoe and Carroll, 1997), (Korhonen, 2002), (Sarkar and Zeman, 2000) and specific biomedical domain corpora as well (Rimell et al., 2013). Such works are limited to the extraction of syntactic frames representing subcategorization information. Instead, we focus on relating the syntactic and semantic frame and, in particular, on the linking between syntactic and semantic arguments.

Another trend of work relevant to this paper is generation from databases using parallel corpora of data and text. (Angeli et al., 2010) train a sequence of discriminative models to predict data selection, ordering and realisation. (Wong and Mooney, 2007) uses techniques from statistical machine translation to model the generation task and (Konstas and Lapata, 2012b; Konstas and Lapata, 2012a) learns a probabilistic Context-Free Grammar modelling the structure of the database and of the associated text. Various systems from the KBGEN shared task (Banik et al., 2013) – (Butler et al., 2013), (Gyawali and Gardent, 2013) and (Zarrieβ and Richardson, 2013) perform generation from the same input data source as ours' and use parallel text for supervision. Our approach differs from all these approaches in that it does not require parallel text/data corpora. Also in contrast to the template extraction approaches described in (Kondadadi et al., 2013), (Ell and Harth, 2014) and (Duma and Klein, 2013), we do not succeed in directly matching the input data to surface text in the sentences obtained from non-parallel biomedical texts. Instead, we must extract the subcategorization frame and learn the linking between semantic and syntactic arguments.

3 Methodology

Our goal is to automatically generate natural language verbalisations of the biological event descriptions encoded in KB BIO 101 (Chaudhri et al., 2013) whereby an *event description* is assumed to consist of an event, its arguments and the roles relating each argument to the event. In the KB BIO 101 knowledge base, events are concepts of type EVENT (e.g., RELEASE), arguments are concepts of type ENTITY (e.g., GATED-CHANNEL, VASCULAR-TISSUE, IRON) and roles are relations between events and entities (e.g., AGENT, PATIENT, PATH, INSTRUMENT).

We propose a probabilistic method which extracts possible verbalisation frames from large biology specific domain corpora and uses probabilities both to select an appropriate frame given an event description and to determine the mapping between syntactic and semantic arguments. That is, probabilities are used to determine which event argument fills which syntactic function (e.g., subject, object) in the produced verbalisation.

We start by giving a brief overview of the content and the structure of KB BIO 101(Section 3.1). We then describe the steps involved in building our generation system.

3.1 KB Bio 101

The foundational component of the KB is the Component Library (CLIB), an upper ontology which is linguistically motivated and designed to support the representation of knowledge for automated reasoning (Gunning et al., 2010). CLIB adopts four simple top level distinctions: (1) enti-

```
SubClassOf(:Hydrophobic-Compound :Entity)
SubClassOf(:Plasma-Membrane :Entity)
SubClassOf(:Block
 ObjectIntersectionOf(:Event
  ObjectSomeValuesFrom(:instrument :Plasma-Membrane)
  ObjectSomeValuesFrom(:object :Hydrophobic-Compound)))
```

Figure 1: Example Event Representation in KB BIO 101

ties (things that are); (2) events (things that happen); (3) relations (associations between things); and (4) roles (ways in which entities participate in events).

Figure 1 shows an example representation for a blocking event between a plasma membrane and hydrophobic compounds which could be verbalised as *The plasma membrane blocks hydrophobic compounds*. In this representation, *Block* is a subclass of the event class. *Plasma-Menbrane* and *Hydrophobic-Compound* are subclasses of the entity class. The *Plasma-Menbrane* and the *Hydrophobic-Compound* concepts stand respectively in an *instrument* and in an *object* role relation with the *Block* event.

KB BIO 101 is organized into a set of concept maps, where each concept map corresponds to a biological entity or process. It was encoded by biology teachers and contains around 5,000 concept maps. KB BIO 101 is available for download for academic purposes in various formats including OWL[2] .

To test and evaluate our approach, we focus on the subpart of KB BIO 101 isolated for the KBGEN surface realisation shared task by (Banik et al., 2013). In this dataset, content units were semi-automatically selected from KB BIO 101 in such a way that (i) the set of relations in each content unit forms a connected graph; (ii) each content unit can be verbalised by a single, possibly complex sentence which is grammatical and meaningful and (iii) the set of content units contain as many different relations and concepts of different semantic types (events, entities, properties, etc) as possible.

That is, the KB content extracted for KBGEN isolate event descriptions which can be verbalised by a single, coherent sentence. To evaluate the ability of our generator to generate event descriptions, we further process this dataset to produce all KB fragments which represent a single event with roles to entities only. The statistics for the resulting dataset (dubbed KBGEN+) are shown in Table 1.

[2]http://www.ai.sri.com/halo/halobook2010/exported-kb/biokb.html

Items	Count
Total nb of Event Descriptions	336
Avg (min/max) nb of roles in an Event Description	2.93/1.8
Total nb of events	126 (336)
Total nb of entities	271 (929)
Total nb of roles	14 (929)

Table 1: Test Set. The numbers in brackets indicate the number of tokens in KBGEN$^+$

3.2 Corpus Collection

We begin by gathering sentences from several of the publicly available biomedical domain corpora.[3] This includes the BioCause (Mihil et al., 2013), BioDef[4], BioInfer (Pyysalo et al., 2007), Grec (Thompson et al., 2009), Genia (Kim et al., 2003) and PubMedCentral (PMC)[5] corpus. We also include the sentences available in annotations of named concepts in the KB BIO 101 ontology. This custom collection of sentences will be the corpus upon which our learning approach will build on. Table 2 lists the count of sentences available in each corpus and in total.

	#Sentences
BioCause	3,187
BioDef	8,426
BioInfer	1,100
Genia	37,092,000
Grec	2,035
PMC	7,018,743
KBBio101	3,393
Total	44,128,884

Table 2: Corpus Size

3.3 Lexicon Creation

To identify corpus sentences which might contain verbalisations of KBGEN$^+$ events and entities, we build a lexicon mapping events and entities contained in KBGEN$^+$ to natural language words or phrases using existing resources. First, we take the lexicon provided by the KBGEN

[3]Ideally, since KB BIO 101 was developed based on a textbook, we would use this textbook as a corpus. Unfortunately, the textbook, previously licensed from Pearson, is no longer available.

[4]Obtained by parsing the ⟨Supplement⟩ section of html pages crawled from http://www.biology-online.org/dictionary/

[5]ftp://ftp.ncbi.nlm.nih.gov/pub/pmc

challenge. The KBGEN lexicon is composed of entries that provide inflected forms and nominalizations for the event variables and singular and plural noun forms for the entity variables, such as :

Block, blocks, block, blocked, blocking
Earthworm, earthworm, earthworms

To this, we add the synset entries of Mesh[6] and the BioDef[7] vocabularies containing the KBGEN$^+$ events and entities . Some example synsets obtained from Mesh and BioDef are shown below:

Block, prevent, stop
Neoplasm,Tumors,Neoplasia,Cancer

Finally, for generalisation purposes, we automatically extract the direct parent and siblings of the KBGEN$^+$ events and entities in the KB BIO 101 ontology and add them as a lexical entries for the corresponding KBGEN$^+$ event/entity. For example, for the KBGEN$^+$ event **"Block"**, the direct parent and siblings extracted from the KB BIO 101 are, respectively:

make inaccessible
conceal, deactivate, obstruct

Our lexicon is then a merge of all entries extracted from either a lexicon or the ontology for the KBGEN$^+$ events and entities. In Table 3, we present the size of lexicon available from each source (Total Entries) and the count of KBGEN$^+$ event and entity types (Intersecting Entries) for which one or more entry was found in that source. Table 4 shows the proportion of KBGEN$^+$ event and entity types for which a lexical entry was found as well as the maximum, minimum and average number of lexical items associated with event and entities in the merged lexicon.

3.4 Frame Extraction

Events in KBGEN$^+$ take an arbitrary number of participants ranging from 1 to 8. Knowing the lexicalisation of an event name is therefore not sufficient. For each event lexicalisation, information about syntactic subcategorisation and syntactic/semantic linking is also required. Consider for instance, the following event representation:

```
SubClassOf(:PC/EBP beta :Entity)
SubClassOf(:TNF-activation :Entity)
SubClassOf(:Myeloid-Cells :Entity)
SubClassOf(:Block
 ObjectIntersectionOf(:Event
  ObjectSomeValuesFrom(:instrument :C/EBP beta)
  ObjectSomeValuesFrom(:object :TNF-activation)))
  ObjectSomeValuesFrom(:base :Myeloid-Cells)))
```

	Total Entries	Intersecting Entries
KBGen	469	397
Mesh	26795	65
BioDef	14934	99
KBBio101	6972	397

Table 3: Lexical Entries and Number of KBGEN$^+$ event and entities for which one or more entry was found (Intersecting Entries)

Knowing that a possible lexicalisation of a *Block* event is the finite verb form *blocked* is not sufficient to produce an appropriate verbalisation of the KB event e.g.,

(1) *C/EBP beta blocked TNF activation in myeloid cells.*

In addition, one must know that this verb (i) takes a subject, an object and an optional prepositional argument introduced by a locative preposition (subcategorisation information) and (ii) that the INSTRUMENT role is realised by the subject slot, the OBJECT role by the DOBJ slot and the BASE role by the PREP-LOC slot (syntax/semantics linking information). That is, we need to know, for each KB event e and its associated roles (i.e., event-to-entity relations), first, what are the syntactic arguments of each possible lexicalisations of e and second, for each possible lexicalisation, which role maps to which syntactic function.

To address this issue, we extract syntactic frames from our constructed corpus and use the collected data to learn the mapping between KB and syntactic arguments.

Frame extraction proceeds as follows. For each event name in the KBGEN$^+$event set, we look for sentences in the corpus that mention this event name or one of its several verbalisations available in the merged lexicon (ALL in Table 4).

We then parse these sentences using the Stanford dependency parser[8] for collapsed dependency structures and extract frames from the resulting

[6] http://www.nlm.nih.gov/mesh/filelist.html

[7] Obtained by parsing the entries in ⟨Synonyms⟩ section of html pages crawled from http://www.biology-online.org/dictionary/

[8] http://nlp.stanford.edu/software/lex-parser.shtml

	KBGen	Mesh	BioDef	KBBio101	ALL	Min/MAx/Avg
Event	100%	10.31%	25.39%	100%	100%	5/97/22
Entity	100%	19.18%	24.72%	100%	100%	3/91/16.18
All	100%	16.37%	24.93%	100%	100%	3/97/18.03

Table 4: Proportion of Event and Entity Names for which a Lexical Entry was found. Min, max and average number of lexical items associated with event and entities

parse trees. A frame is a sequence of dependency relations labelling the local subtree originating at a node labelled with an event name (or one of its variants). For instance, given the sentence and the dependency tree shown in Figure 2, the extracted frame for the event *Block* will be :

nsubj,VB,dobj

indicating that the verb form *block* requires a subject and an object.

That is, a syntactic frame describes the arguments required by the lexicalisations of an event and the syntactic function they realise.

When extracting the frames, we only consider a subset of the dependency relations produced by the Stanford parser to avoid including in the frame adjuncts such as temporal or spatial phrases which are optional rather than required arguments. Specifically, the dependency relations considered for frame construction are:

agent, amod, dobj, nsubj, nsubjpass, prep_across, prep_along, prep_at, prep_away_from, prep_down, prep_for, prep_from, prep_in, prep_inside, prep_into, prep_of, prep_out_of, prep_through, prep_to, prep_toward, prep_towards, prep_via, prep_with, vmod_creating, vmod_forming, vmod_producing, vmod_resulting, vmod_using, xcomp_using, auxpass.[9]

A total of 718 distinct event frames were observed whereby 97.63% of the KBGEN$^+$events were assigned at least one frame and each event was assigned an average of 82.01 distinct frames. Each event can be lexicalised by several natural language words or phrases and each natural language expressions may occur in several syntactic environments.

[9] *vmod_creating, vmod_forming, vmod_producing, vmod_resulting, vmod_using, xcomp_using* are not directly given by the Stanford parser but reconstructed from a *vmod* or an *xcomp* dependency "collapsed" with the lemmas *producing* or *using* much in the same way as the *prep_P* collapsed dependency relation provided by the Stanford Parser. These added dependencies are often used in biomedical text to express e.g., RESULT or RAW-MATERIAL role relations.

3.5 Probabilistic Models

Given F a set of syntactic frames, E a set of KB event names, D a set of syntactic dependency names and R, a set of KB roles, we next describe three probabilistic models that will be used to generate natural language sentences.

- The model $P(f|e)$ with $f \in F$ and $e \in E$, which encodes the probability of a frame given an event.
- The model $P(f|r)$ with $f \in F$ and $r \in R$, which encodes the probability of a frame given a role.
- The model $P(d|r)$ with $d \in D$ and $r \in R$, which encodes the probability of a syntactic dependency given a role.

We have chosen generative models for frames and dependencies given events and roles, and not the other way around, because such models intuitively match the generation process at test time. Each of the three models $P(f|e)$, $P(f|r)$ and $P(d|r)$ is assumed multinomial with maximum likelihood estimates determined by the labelled data built as described in Algorithm 1. Intuitively, C_e is the corpus consisting of all frames found in the corpus to be associated with a lexicalisation of e. Similarly, C_r nd C_d gathers all pairs of (frame,role) and (dependency relation, role) that could be identified given the KBGEN$^+$ KB, the corpus described in Section 3.2 and the lexicon described in Section 3.3. A Symmetric Dirichlet prior with hyperparameter $\alpha = 0.1$ is further used in order to favor sparse distributions. Training thus gives:

$$P(f|e) = \frac{counts\left((f,e) \in \mathcal{C}_e\right) + 0.1}{\sum_{f'}\left(counts\left((f',e) \in \mathcal{C}_e\right) + 0.1\right)}$$

This first model allows to choose a syntactic frame that will be used to verbalize a given event.

For the second distribution:

$$P(f|r) = \frac{counts\left((f,r) \in \mathcal{C}_r\right) + 0.1}{\sum_{f'}\left(counts\left((f',r) \in \mathcal{C}_r\right) + 0.1\right)}$$

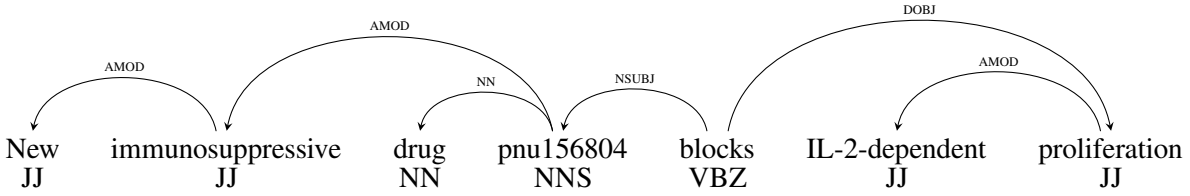

Figure 2: Example Dependency Parse Tree

This second model also ranks the frames, but this time based on the given set of roles.

The third model is trained in a similar way:

$$P(d|r) = \frac{counts\left((d,r) \in \mathcal{C}_d\right) + 0.1}{\sum_{d'}\left(counts\left((d',r) \in \mathcal{C}_d\right) + 0.1\right)}$$

It is used to choose which dependency in f shall represent the role r.

3.6 Surface Realisation

In our approach, surface realisation takes as input an even description. To verbalize an input event description containing an event e and n roles $r_1 \ldots\ldots r_n$, we first identify the event and the roles present in the input. The arity of the event is then defined as the count of distinct role types present in the input (to favor aggregation, in case of repeating roles)[10]. Among all the frames seen for this event during training, we select only those that have the same arity (same number of syntactic dependents) as the input event. All such frames are candidate frames for generation.

We consider two alternative scoring functions for choosing the n-best frames[11]. In the first case, we select the frame which maximises the score (M1):

$$P(f|e) \times \prod_{i=1}^{n} P(f|r_i) \qquad \text{(M1)}$$

To determine the mapping between roles and syntactic dependencies, we then look for the best permutation of the roles for every winning frame $f = (d_1, \cdots, d_n)$:

$$(\hat{r}_1^f, \ldots, \hat{r}_n^f) = \underset{(r_1,\ldots,r_n)\in\mathcal{P}(\{r_1,\ldots,r_n\})}{\arg\max} \prod_{i=1}^{n} P(d_i|r_i)$$

where $\mathcal{P}(\{r_1, \ldots, r_n\})$ is the set of all permutations of the roles[12].

In the second model (M2), we first compute the optimal mapping $(\hat{r}_1^f, \ldots, \hat{r}_n^f)$ for every possible frame and then use this information to select the n-best frames for generation:

$$P(f|e) \times \prod_{i=1}^{n} P(f|r_i) \times \prod_{i=1}^{n} P(d_i|\hat{r}_i^f) \qquad \text{(M2)}$$

Note that (M1) (and (M2)) can be viewed as a *product of experts*, but with independently trained experts and without any normalization factor. It is thus not a probability, but this is fine because the normalization term does not impact the choice of the winning frame.

Both (M1) and (M2) alternatives output a winning $\hat{f}$, i.e., a sequence of dependencies that shall be used to generate the sentence, as well as their mapping with roles $(\hat{r}_1^{\hat{f}}, \ldots, \hat{r}_n^{\hat{f}})$. Thus, generation boils down to filling every dependency slot in sequence with its optional preposition (e.g., for d_i = prep_to or d_i = prep_at) and the lexical entry of the entity bound to the corresponding role. For repeating roles of the input, we aggregate their bound entities via the conjunction "and" and fill the corresponding dependency slot.

The results obtained by verbalising the n-best frames given by models (M1 & M2) are separately stored and we present their analysis in Section 4.

4 Results and Discussion

We evaluate our approach on the 336 event representations included in the KBGEN$^+$ dataset. For each event representation, we generate the 5 best natural language verbalisations using the method described in the preceding section. We then evaluate the results both qualitatively and quantitatively.

[10]Thus if the input event description contains e.g., 2 object roles and an instrument role, its arity will be 2 rather than 3. This accounts for the fact that the two object roles will be verbalised as a coordinated NP filling in a single dependency function rather than two distinct syntactic arguments.

[11]n=5 in our experiments

[12]Here, we assume the order of dependencies in f is fixed, and we permute the roles; this is of course equivalent to permuting the dependencies with fixed roles sequences.

Input	KBGEN$^+$
	Lexicons $\mathcal{L}_e$ for events and $\mathcal{L}_t$ for entities as described in Section 3.3
	Raw text corpus $\mathcal{T}$ with dependency trees as described in Section 3.4
Output	Corpus (multiset) $\mathcal{C}_e$ for model $P(f\|e)$
	Corpus (multiset) $\mathcal{C}_r$ for model $P(f\|r)$
	Corpus (multiset) $\mathcal{C}_d$ for model $P(d\|r)$

1. For every event $e \in$KBGEN$^+$, let $lex(e)$ be all possible lexicalisations of e taken from $\mathcal{L}_e$:
2. For every lexicalisation $l \in lex(e)$:
3. For every occurrence $e_t \in \mathcal{T}$ of l:
   (a) Extract the frame f governed by e_t
   (b) Add the observation f with label e in the frame-event corpus:

$$\mathcal{C}_e \leftarrow \mathcal{C}_e \uplus \{\{(f, e)\}\}$$

   (c) For every entity $w_t \in \mathcal{L}_t$ that is a dependent of e_t with syntactic relation d, add every role r associated with this entity in KBGEN$^+$ to both role corpora:

$$\mathcal{C}_r \leftarrow \mathcal{C}_r \uplus \{\{(f, r)\}\}$$

$$\mathcal{C}_d \leftarrow \mathcal{C}_d \uplus \{\{(d, r)\}\}$$

Algorithm 1: Preparation of the corpora used to train our probabilistic models

4.1 Coverage

We first consider coverage i.e., the proportion of input in the test set for which a verbalisation is produced. In total, we generate output for 321 (95.53%) of the test data.

For 3 input cases involving two distinct event names (PHOTORESPIRATION, UNEQUAL-SHARING), there was no associated frame because none of the lexicalisations of the event name could be found in the corpus. Covering such cases would involve a more sophisticated lexicalisation strategy for instance, the strategy used in (Trevisan, 2010), where names are tokenized and pos-tagged before being mapped using hand-written rules to a lexicalisation.

For the other 12 input cases, generation fails because no frame of matching arity could be found. As discussed in Section 4.3 below, this is often due to cases where a KB role (mostly the BASE role) is verbalised as a modifier of an argument rather than a verb argument. Other cases are cases where the event is nominalised and there is no matching frame for that nominalisation.

4.2 Accuracy

Because the generated verbalisations are not learned from a parallel corpora, the generated sentences are often very different from the reference sentence. For instance, the generated sentence may contain a verb while in the reference sentence, the event is nominalised. Or the event might be verbalised by a transitive verb in the generated sentence but by a verb taking a prepositional object in the reference sentence (Eg: *A double bond holds together an oxygen and a carbon* vs. *Carbon and oxygen are held together by double bond*). To automatically assess the quality of the generated sentences, we therefore do not use BLEU. Instead we measure the accuracy of role mapping and we complement this automatic metric with the human evaluation described in the next section.

Role mapping is assessed as follows. For each input in the test data, we record the mapping beween the KB role of an argument in the event description and the syntactic dependency of the corresponding natural language argument in the gold sentence. For instance, given the event description shown in Section 3.4 for Sentence 1 (repeated below for convinience as Example 1), we record the syntax/semantics mapping: INSTRUMENT:NSUBJ, OBJECT:DOBJ, BASE:PREP-IN.

Example 1

```
SubClassOf(:PC/EBP beta :Entity)
SubClassOf(:TNF-activation :Entity)
SubClassOf(:Myeloid-Cells :Entity)
SubClassOf(:Block
 ObjectIntersectionOf(:Event
  ObjectSomeValuesFrom(:instrument :C/EBP beta)
  ObjectSomeValuesFrom(:object :TNF-activation)))
  ObjectSomeValuesFrom(:base :Myeloid-Cells)))
```

C/EBP beta blocked TNF activation in myeloid cells.

Accuracy is then the proportion of generated role:dependency mappings which are correct i.e., which match the reference. Although this does not address the fact that the generated and the reference sentence may be very different, it provides some indication of whether the generated mappings are plausible. We thus report this accuracy for the 1-best and 5-best solutions provided by our model, to partly account for the variability in possible correct answers. We

compare our results to two baselines. The first baseline (BL-LING) is obtained using a default role/dependency assignment which is manually defined using linguistic introspection. The second (BL-GOLD) is a strong, informed baseline which has access to the frequency of the role/dependency mapping in the gold corpus. That is, this second baseline assigns to each role in the input event description, the syntactic dependency most frequently assigned to this role in the gold corpus. The default mapping used for BL-GOLD is as follows: *toward/prep_towards, site/prep_in, result/dobj, recipient/prep_to, raw_material/dobj, path/prep_through, origin/prep_from, object/dobj, instrument/nsubj, donor/prep_from, destination/prep_into, base/prep_in, away-from/prep_away_from, agent/nsubj*. The manually defined mapping used for BL-LING differs on three mappings namely *raw_material/prep-from, instrument-with,destination-to*.

On the 336 event descriptions (929 roles occurrences) contained in the test set, we obtain the following results:

Scoring	5-best acc	1-best acc
BL-Ling		42%
BL-GOLD		49%
M1	48%	30%
M2	49%	31%
M2-BL-LING	57%	43%

As expected, the difference between BL-LING and BL-GOLD shows that using information from the GOLD strongly improves accuracy.

While M1 and M2 do not improve on the baseline, an important drawback of these baselines is that they may map two or more roles in an event description to the same dependency (e.g., RAW-MATERIAL and RESULT to *dobj*). Worse, they may map a role to a dependency which is absent from the selected frame (if the dependency mapped onto by a role in the input does not exist in that frame). In contrast, the probabilistic approach is linguistically more promising as it guarantees that each role is mapped to a distinct dependency relation. We therefore take advantage of both the linguistically inspired baseline (BL-LING) and the probabilistic approach by combining both into a model (M2-BL-LING) which simply replaces the mapping proposed by the M2 model by that proposed by the BL-LING baseline whenever the probability of the M2 model is below a given threshold[13]. Because it predicts role/dependency mappings that are consistent with the selected frames, this new model is linguistically sound. And because it makes use of the strong prior information contained in the BL-LING baseline, it has a good accuracy.

4.3 Human Evaluation

Taking a sample of 264 inputs from the KBGEN$^+$ dataset, we evaluate the mappings of roles to syntax in the output. The sample contains inputs with 1 to 2 roles (40%), 3 roles (30%) and more than 3 roles (30%). For each sampled input, we consider the 5 best outputs and manually grade the output as follows:

1. Correct: both the syntax/semantic linking of the arguments and the lexicalisation of the event and of its arguments is correct.

2. Almost Corrrect: the lexicalisation of the event and of its arguments is correct and the linking of core semantic arguments is correct. The core arguments are the most frequent ones in the test data namely AGENT, BASE, OBJECT.

3. Incorrect: all other cases.

Three judges independently graded 264 inputs using the above criteria. The inter-annotator agreement, as measured with the Fleiss Kappa in a preliminary experiment in these conditions, was $\kappa = 0.76$ which is considered as "good agreement" in the literature. 29% of the ouput were found to be correct, 20% to be almost correct and 51% to be incorrect.

One main factor negatively affecting results is the number of roles contained in an event description. Unsurprisingly, the greater the number of roles the lower the accuracy. That is, for event descriptions with 3 or less roles, the scores are higher (40%, 23%, 37% respectively for correct, almost correct and incorrect) as there are less possibilities to be considered. Another, related issue, is data sparsity. Unsurprisingly, roles that are less frequent often score lower (i.e., are more often incorrectly mapped to syntax) than roles which occur more frequently. Thus, the three most frequent roles (AGENT,OBJECT, BASE) have a 5-best role mapping accuracy that ranges from 43% to 77%, while most other roles have much lower accuracy. These

[13]We have empirically chosen a threshold that retains 40% of our model's outputs; this is the only threshold value that we have tried, and we have not tuned this threshold at all

two issues suggest that results could be improved by using either more data or a more sophisticated smoothing or learning strategy. However linguistic factors are also at play here.

First, some semantic roles are often verbalised as verbs rather than thematic roles. For instance, in Sentence (2), the event (INTRACELLULAR-DIGESTION) is verbalised as a nominalisation and the OBJECT role as a verb (*produces*). More generally, a role in the KB is not necessarily realised by a thematic role.

(2) Intracellular digestion of polymers and solid substances in the lysosome produces monomers.

Second, in some cases, entities which are arguments of the event in the input are verbalised as prepositional modifiers of an argument of the verb verbalising the event rather than as an argument of the verb itself. This is frequently the case for the BASE relation. For instance, Example (3) shows the gold sentence for an input containing EUKARYOTIC-CELL as a BASE argument. As can be seen, in this case, the EUKARYOTIC-CELL entity is verbalised by a prepositional phrase modifying an NP rather than by an argument of the verb.

(3) Lysosomal enzymes digest nucleic acids and proteins in the lysosome of eukaryotic cells.

5 Conclusion

We have presented an approach for verbalising biological event representations which differs from previous work in that (i) it uses a non-parallel corpora and (ii) it focuses on n-ary relations and on the issue of how to automatically map natural language and KB arguments. A first evaluation gives encouraging results and identifies three main open questions for further research. How best to deal with data sparsity to account for event descriptions involving a high number of roles or roles that are infrequent? How to handle semantic roles that are verbalised as modifiers rather than as syntactic arguments? How to account for cases where KB roles are verbalised by verbs rather than by syntactic dependencies?

References

G. Aguado, A. Bañón, J. Bateman, S. Bernardos, M. Fernández, A. Gómez-Pérez, E. Nieto, A. Olalla, R. Plaza, and A. Sánchez. 1998. Ontogeneration: Reusing domain and linguistic ontologies for spanish text generation. In *Workshop on Applications of Ontologies and Problem Solving Methods, ECAI*, volume 98.

Gabor Angeli, Percy Liang, and Dan Klein. 2010. A simple domain-independent probabilistic approach to generation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 502–512. Association for Computational Linguistics.

Eva Banik, Claire Gardent, and Eric Kow. 2013. The kbgen challenge. In *the 14th European Workshop on Natural Language Generation (ENLG)*, pages 94–97.

Anja Belz. 2008. Automatic generation of weather forecast texts using comprehensive probabilistic generation-space models. *Natural Language Engineering*, 14(4):431–455.

K. Bontcheva and Y. Wilks. 2004. Automatic report generation from ontologies: the miakt approach. In *Ninth International Conference on Applications of Natural Language to Information Systems (NLDB'2004)*. Lecture Notes in Computer Science 3136, Springer, Manchester, UK.

Ted Briscoe and John Carroll. 1997. Automatic extraction of subcategorization from corpora. In *Proceedings of the fifth conference on Applied natural language processing*, pages 356–363. Association for Computational Linguistics.

Keith Butler, Priscilla Moraes, Ian Tabolt, and Kathy McCoy. 2013. Team udel kbgen 2013 challenge. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 206–207, Sofia, Bulgaria, August. Association for Computational Linguistics.

Vinay K Chaudhri, Michael A Wessel, and Stijn Heymans. 2013. Kb_bio_101: A challenge for owl reasoners. In *ORE*, pages 114–120. Citeseer.

David L Chen and Raymond J Mooney. 2008. Learning to sportscast: a test of grounded language acquisition. In *Proceedings of the 25th international conference on Machine learning*, pages 128–135. ACM.

Deborah A Dahl, Madeleine Bates, Michael Brown, William Fisher, Kate Hunicke-Smith, David Pallett, Christine Pao, Alexander Rudnicky, and Elizabeth Shriberg. 1994. Expanding the scope of the atis task: The atis-3 corpus. In *Proceedings of the workshop on Human Language Technology*, pages 43–48. Association for Computational Linguistics.

Daniel Duma and Ewan Klein, 2013. *Generating Natural Language from Linked Data: Unsupervised Template Extraction*, pages 83–94. ASSOC COMPUTATIONAL LINGUISTICS-ACL.

Basil Ell and Andreas Harth. 2014. A language-independent method for the extraction of rdf verbalization templates. *INLG 2014*, page 26.

D. Galanis, G. Karakatsiotis, G. Lampouras, and I. Androutsopoulos. 2009. An open-source natural language generator for owl ontologies and its use in protégé and second life. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics: Demonstrations Session*, pages 17–20. Association for Computational Linguistics.

David Gunning, Vinay K Chaudhri, Peter E Clark, Ken Barker, Shaw-Yi Chaw, Mark Greaves, Benjamin Grosof, Alice Leung, David D McDonald, Sunil Mishra, et al. 2010. Project halo updateprogress toward digital aristotle. *AI Magazine*, 31(3):33–58.

Bikash Gyawali and Claire Gardent. 2013. Lor-kbgen, a hybrid approach to generating from the kbgen knowledge-base. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 204–205, Sofia, Bulgaria, August. Association for Computational Linguistics.

K. Kaljurand and N.E. Fuchs. 2007. Verbalizing owl in attempto controlled english. *Proceedings of OWLED07*.

J.-D. Kim, T. Ohta, Y. Tateisi, and J. Tsujii. 2003. Genia corpusa semantically annotated corpus for biotextmining. *Bioinformatics*, 19(suppl 1):i180–i182.

Ravi Kondadadi, Blake Howald, and Frank Schilder. 2013. A statistical nlg framework for aggregated planning and realization.

Ioannis Konstas and Mirella Lapata. 2012a. Concept-to-text generation via discriminative reranking. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 369–378. Association for Computational Linguistics.

Ioannis Konstas and Mirella Lapata. 2012b. Unsupervised concept-to-text generation with hypergraphs. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 752–761. Association for Computational Linguistics.

Anna Korhonen. 2002. Semantically motivated subcategorization acquisition. In *Proceedings of the ACL-02 Workshop on Unsupervised Lexical Acquisition - Volume 9*, ULA '02, pages 51–58, Stroudsburg, PA, USA. Association for Computational Linguistics.

Percy Liang, Michael I Jordan, and Dan Klein. 2009. Learning semantic correspondences with less supervision. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 91–99. Association for Computational Linguistics.

C. Mihil, T. Ohta, S. Pyysalo, and S. Ananiadou. 2013. Biocause: Annotating and analysing causality in the biomedical domain. *BMC BioInformatics*.

VO Mittal, G. Carenini, and JD Moore. 1994. Generating patient specific explanations in migraine. In *Proceedings of the eighteenth annual symposium on computer applications in medical care. McGraw-Hill Inc*.

C.L. Paris. 1988. Tailoring object descriptions to a user's level of expertise. *Computational Linguistics*, 14(3):64–78.

Sampo Pyysalo, Filip Ginter, Juho Heimonen, Jari Bjrne, Jorma Boberg, Jouni Jrvinen, and Tapio Salakoski. 2007. Bioinfer: A corpus for information extraction in the biomedical domain. *BMC Bioinformatics*.

E. Reiter, R. Robertson, and L.M. Osman. 2003. Lessons from a failure: Generating tailored smoking cessation letters. *Artificial Intelligence*, 144(1):41–58.

Laura Rimell, Thomas Lippincott, Karin Verspoor, Helen L Johnson, and Anna Korhonen. 2013. Acquisition and evaluation of verb subcategorization resources for biomedicine. *Journal of biomedical informatics*, 46(2):228–237.

Anoop Sarkar and Daniel Zeman. 2000. Automatic extraction of subcategorization frames for czech. In *Proceedings of the 18th conference on Computational linguistics-Volume 2*, pages 691–697. Association for Computational Linguistics.

P. Thompson, S. A. Iqbal, J. McNaught, and S. Ananiadou. 2009. Construction of an annotated corpus to support biomedical information extraction. *BMC Bioinformatics*.

Marco Trevisan. 2010. A portable menuguided natural language interface to knowledge bases for querytool. Master's thesis, Free University of Bozen-Bolzano (Italy) and University of Groningen (Netherlands).

G. Wilcock. 2003. Talking owls: Towards an ontology verbalizer. *Human Language Technology for the Semantic Web and Web Services, ISWC*, 3:109–112.

Sandra Williams and Richard Power. 2010. Grouping axioms for more coherent ontology descriptions. In *Proceedings of the 6th International Natural Language Generation Conference (INLG 2010)*, pages 197–202, Dublin.

Yuk Wah Wong and Raymond J Mooney. 2007. Generation by inverting a semantic parser that uses statistical machine translation. In *HLT-NAACL*, pages 172–179.

Sina Zarrieβ and Kyle Richardson. 2013. An automatic method for building a data-to-text generator. In *Proceedings of the 14th European Workshop on Natural Language Generation*, Sofia, Bulgaria, August.