Surface Realisation from Knowledge-Bases

Bikash Gyawali
Université de Lorraine, LORIA
Villers-lès-Nancy, F-54600, France
bikash.gyawali@loria.fr

Claire Gardent
CNRS, LORIA, UMR 7503
Vandoeuvre-lès-Nancy, F-54500, France
claire.gardent@loria.fr

Abstract

We present a simple, data-driven approach to generation from knowledge bases (KB). A key feature of this approach is that grammar induction is driven by the extended domain of locality principle of TAG (Tree Adjoining Grammar); and that it takes into account both syntactic and semantic information. The resulting extracted TAG includes a unification based semantics and can be used by an existing surface realiser to generate sentences from KB data. Experimental evaluation on the KBGen data shows that our model outperforms a data-driven generate-and-rank approach based on an automatically induced probabilistic grammar; and is comparable with a handcrafted symbolic approach.

1 Introduction

In this paper we present a grammar based approach for generating from knowledge bases (KB) which is linguistically principled and conceptually simple. A key feature of this approach is that grammar induction is driven by the extended domain of locality principle of TAG (Tree Adjoining Grammar) and takes into account both syntactic and semantic information. The resulting extracted TAGs include a unification based semantics and can be used by an existing surface realiser to generate sentences from KB data.

To evaluate our approach, we use the benchmark provided by the KBGen challenge (Banik et al., 2012; Banik et al., 2013), a challenge designed to evaluate generation from knowledge bases; where the input is a KB subset; and where the expected output is a complex sentence conveying the meaning represented by the input. When compared with two other systems having taken part in the KBGen challenge, our system outperforms a data-driven, generate-and-rank approach based on an automatically induced probabilistic grammar; and produces results comparable to those obtained by a symbolic, rule based approach. Most importantly, we obtain these results using a general purpose approach that we believe is simpler and more transparent than current state of the art surface realisation systems generating from KB or DB data.

2 Related Work

Our work is related to work on concept to text generation.

Earlier work on concept to text generation mainly focuses on generation from logical forms using rule-based methods. (Wang, 1980) uses hand-written rules to generate sentences from an extended predicate logic formalism; (Shieber et al., 1990) introduces a head-driven algorithm for generating from logical forms; (Kay, 1996) defines a chart based algorithm which enhances efficiency by minimising the number of semantically incomplete phrases being built; and (Shemtov, 1996) presents an extension of the chart based generation algorithm presented in (Kay, 1996) which supports the generation of multiple paraphrases from underspecified semantic input. In all these approaches, grammar and lexicon are developed manually and it is assumed that the lexicon associates semantic sub-formulae with natural language expressions. Our approach is similar to these approaches in that it assumes a grammar encoding a compositional semantics. It differs from them however in that, in our approach, grammar and lexicon are automatically acquired from the data.

With the development of the semantic web and the proliferation of knowledge bases, generation from knowledge bases has attracted increased interest and so called ontology verbalisers have been proposed which support the generation of text from (parts of) knowledge bases. One main

Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pages 424–434,
Baltimore, Maryland, USA, June 23-25 2014.

strand of work maps each axiom in the knowledge base to a clause. Thus the OWL verbaliser integrated in the Protégé tool (Kaljurand and Fuchs, 2007) provides a verbalisation of every axiom present in the ontology under consideration and (Wilcock, 2003) describes an ontology verbaliser using XML-based generation. As discussed in (Power and Third, 2010), one important limitation of these approaches is that they assume a simple deterministic mapping between knowledge representation languages and some controlled natural language (CNL). Specifically, the assumption is that each atomic term (individual, class, property) maps to a word and each axiom maps to a sentence. As a result, the verbalisation of larger ontology parts can produce very unnatural text such as, *Every cat is an animal. Every dog is an animal. Every horse is an animal. Every rabbit is an animal.* More generally, the CNL based approaches to ontology verbalisation generate clauses (one per axiom) rather than complex sentences and thus cannot adequately handle the verbalisation of more complex input such as the KBGen data where the KB input often requires the generation of a complex sentence rather than a sequence of base clauses.

To generate more complex output from KB data, several alternative approaches have been proposed.

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. This approach however relies on extensive manual annotation of the input data.

The SWAT project has focused on producing descriptions of ontologies that are both coherent and efficient (Williams and Power, 2010). For instance, instead of the above output, the SWAT system would generate the sentence: *The following are kinds of animals: cats, dogs, horses and rabbits.* . In this approach too however, the verbaliser output is strongly constrained by a simple Definite Clause Grammar covering simple clauses and sentences verbalising aggregation patterns such as the above. More generally, the sentences generated by ontology verbalisers cover a limited set of linguistics constructions; the grammar used is manually defined; and the mapping between semantics and strings is assumed to be deterministic (e.g., a verb maps to a relation and a noun to a concept). In constrast, we propose an approach which can generate complex sentences from KB data; where the grammar is acquired from the data; and where no assumption is made about the mapping between semantics and NL expressions.

Recent work has focused on data-driven generation from frames, lambda terms and data base entries.

(DeVault et al., 2008) describes an approach for generating from the frames produced by a dialog system. They induce a probabilistic Tree Adjoining Grammar from a training set aligning frames and sentences using the grammar induction technique of (Chiang, 2000) and use a beam search that uses weighted features learned from the training data to rank alternative expansions at each step.

(Lu and Ng, 2011) focuses on generating natural language sentences from logical form (i.e., lambda terms) using a synchronous context-free grammar. They introduce a novel synchronous context free grammar formalism for generating from lambda terms; induce such a synchronous grammar using a generative model; and extract the best output sentence from the generated forest using a log linear model.

(Wong and Mooney, 2007; Lu et al., 2009) focuses on generating from variable-free tree-structured representations such as the CLANG formal language used in the ROBOCUP competition and the database entries collected by (Liang et al., 2009) for weather forecast generation and for the air travel domain (ATIS dataset) by (Dahl et al., 1994). (Wong and Mooney, 2007) uses synchronous grammars to transform a variable free tree structured meaning representation into sentences. (Lu et al., 2009) uses a Conditional Ran-

The function of a gated channel is to release particles from the endoplasmic reticulum

```
:TRIPLES (
(|Release-Of-Calcium646| |object| |Particle-In-Motion64582|)
(|Release-Of-Calcium646| |base| |Endoplasmic-Reticulum64603|)
(|Gated-Channel64605|  |has-function||Release-Of-Calcium646|)
(|Release-Of-Calcium646|  |agent| |Gated-Channel64605|))
:INSTANCE-TYPES
(|Particle-In-Motion64582| |instance-of| |Particle-In-Motion|)
(|Endoplasmic-Reticulum64603| |instance-of| |Endoplasmic-Reticulum|)
(|Gated-Channel64605| |instance-of| |Gated-Channel|)
 |Release-Of-Calcium646| |instance-of| |Release-Of-Calcium|))
:ROOT-TYPES (
(|Release-Of-Calcium646| |instance-of| |Event|)
(|Particle-In-Motion64582| |instance-of| |Entity|)
(|Endoplasmic-Reticulum64603| |instance-of| |Entity|)
(|Gated-Channel64605| |instance-of| |Entity|)))
```

Figure 1: Example KBGEN Scenario

dom Field to generate from the same meaning representations.

Finally, more recent papers propose approaches which perform both surface realisation and content selection. (Angeli et al., 2010) proposes a log linear model which decomposes into a sequence of discriminative local decisions. The first classifier determines which records to mention; the second, which fields of these records to select; and the third, which words to use to verbalise the selected fields. (Kim and Mooney, 2010) uses a generative model for content selection and verbalises the selected input using WASP^{-1}, an existing generator. Finally, (Konstas and Lapata, 2012b; Konstas and Lapata, 2012a) develop a joint optimisation approach for content selection and surface realisation using a generic, domain independent probabilistic grammar which captures the structure of the database and the mapping from fields to strings. They intersect the grammar with a language model to improve fluency; use a weighted hypergraph to pack the derivations; and find the best derivation tree using Viterbi algorithm.

Our approach differs from the approaches which assume variable free tree structured representations (Wong and Mooney, 2007; Lu et al., 2009) and data-based entries (Kim and Mooney, 2010; Konstas and Lapata, 2012b; Konstas and Lapata, 2012a) in that it handles graph-based, KB input and assumes a compositional semantics. It is closest to (DeVault et al., 2008) and (Lu and Ng, 2011) who extract a grammar encoding syntax and semantics from frames and lambda terms respectively. It differs from the former however in that it enforces a tighter syntax/semantics integration by requiring that the elementary trees of our extracted grammar encode the appropriate linking information. While (DeVault et al., 2008) extracts a TAG grammar associating each elementary tree with a semantics, we additionnally require that these trees encode the appropriate linking between syntactic and semantic arguments thereby restricting the space of possible tree combinations and drastically reducing the search space. Although conceptually related to (Lu and Ng, 2011), our approach extracts a unification based grammar rather than one with lambda terms. The extraction process and the generation algorithms are also fundamentally different. We use a simple mainly symbolic approach whereas they use a generative approach for grammar induction and a discriminative approach for sentence generation.

3 The KBGen Task

The KBGen task was introduced as a new shared task at Generation Challenges 2013 (Banik et al., 2013)[1] and aimed to compare different generation systems on KB data. Specifically, the task is to verbalise a subset of a knowledge base. For instance, the KB input shown in Figure 1 can be verbalised as:

(1) The function of a gated channel is to release particles from the endoplasmic reticulum

The KB subsets forming the KBGen input data were pre-selected from the AURA biology knowledge base (Gunning et al., 2010), a knowledge base about biology which was manually encoded by biology teachers and encodes knowledge about events, entities, properties and relations where relations include event-to-entity, event-to-event,

[1] http://www.kbgen.org

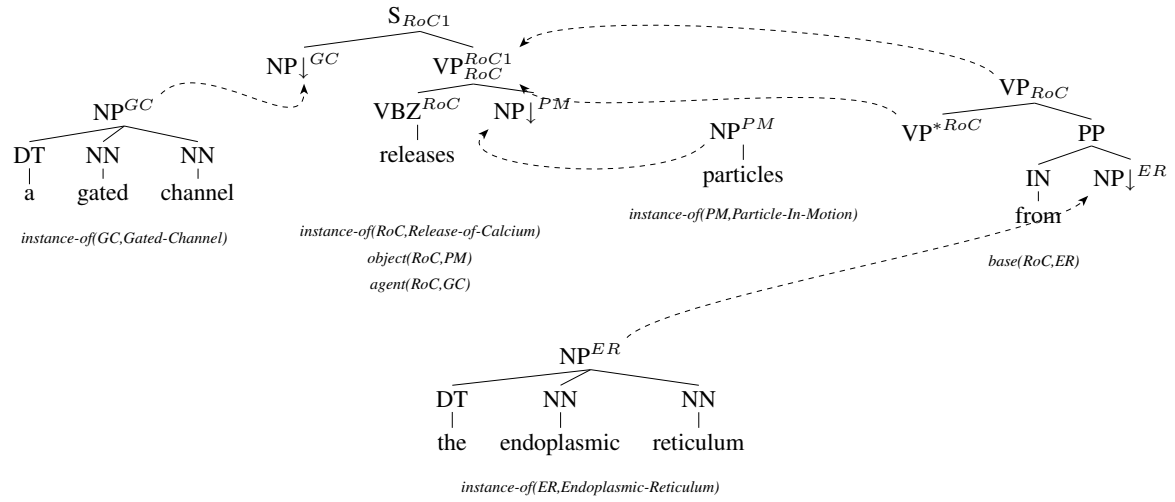

Figure 2: Example FB-LTAG with Unification-Based Semantics. Dotted lines indicate substitution and adjunction operations between trees. The variables decorating the tree nodes (e.g., GC) abbreviate feature structures of the form $[idx : V]$ where V is a unification variable shared with the semantics.

event-to-property and entity-to-property relations. AURA uses a frame-based knowledge representation and reasoning system called Knowledge Machine (Clark and Porter, 1997) which was translated into first-order logic with equality and from there, into multiple different formats including SILK (Grosof, 2012) and OWL2 (Motik et al., 2009). It is available for download in various formats including OWL[2].

4 Generating from the KBGen Knowledge-Base

To generate from the KBGen data, we induce a Feature-Based Lexicalised Tree Adjoining Grammar (FB-LTAG, (Vijay-Shanker and Joshi, 1988)) augmented with a unification-based semantics (Gardent and Kallmeyer, 2003) from the training data. We then use this grammar and an existing surface realiser to generate from the test data.

4.1 Feature-Based Lexicalised Tree Adjoining Grammar

Figure 2 shows an example FB-LTAG augmented with a unification-based semantics.

Briefly, an FB-LTAG consists of a set of elementary trees which can be either initial or auxiliary. Initial trees are trees whose leaves are labeled with substitution nodes (marked with a down-arrow) or terminal categories. Auxiliary trees are distinguished by a foot node (marked with a star) whose category must be the same as that of the root node. In addition, in an FB-LTAG, each elementary tree is anchored by a lexical item (lexicalisation) and the nodes in the elementary trees are decorated with two feature structures called *top* and *bottom* which are unified during derivation. Two tree-composition operations are used to combine trees namely, substitution and adjunction. While substitution inserts a tree in a substitution node of another tree, adjunction inserts an auxiliary tree into a tree. In terms of unifications, substitution unifies the top feature structure of the substitution node with the top feature structure of the root of the tree being substituted in. Adjunction unifies the top feature structure of the root of the tree being adjoined with the top feature structure of the node being adjoined to; and the bottom feature structure of the foot node of the auxiliary tree being adjoined with the bottom feature structure of the node being adjoined to.

In an FB-LTAG augmented with a unification-based semantics, each tree is associated with a semantics i.e., a set of literals whose arguments may be constants or unification variables. The semantics of a derived tree is the union of the semantics of the tree contributing to its derivation modulo unification. Importantly, semantic variables are shared with syntactic variables (i.e., variables occurring in the feature structures decorating the tree nodes) so that when trees are combined, the appropriate syntax/semantics linking is enforced. For instance given the semantics:

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

instance-of(RoC,Release-Of-Calcium),
object(RoC,PM),agent(RoC,GC),base(RoC,ER),
instance-of(ER,Endoplasmic-Reticulum),
instance-of(GC,Gated-Channel),
instance-of(PM,Particle-In-Motion)

the grammar will generate *A gated channel releases particles from the endoplasmic reticulum* but not e.g., *Particles releases a gated channel from the endoplasmic reticulum*.

4.2 Grammar Extraction

We extract our FB-LTAG with unification semantics from the KBGen training data in two main steps. First, we align the KB data with the input string. Second, we induce a Tree Adjoining Grammar augmented with a unification-based semantics from the aligned data.

4.2.1 Alignment

Given a Sentence/Input pair (S, I) provided by the KBGen Challenge, the alignment procedure associates each entity and event variable in I to a substring in S. To do this, we use the entity and the event lexicon provided by the KBGen organiser. The event lexicon maps event types to verbs, their inflected forms and nominalizations while the entity lexicon maps entity types to a noun and its plural form. For instance, the lexicon entries for the event and entity types shown in Figure 1 are as shown in Figure 3.

For each entity and each event variable V in I, we retrieve the corresponding type (e.g., `Particle-In-Motion` for `Particle-In-Motion64582`); search the KBGen lexicon for the corresponding phrases (e.g., *molecule in motion,molecules in motion*); and associate V with the phrase in S which matches one of these phrases. Figure 3 shows an example lexicon and the resulting alignment obtained for the scenario shown in Figure 1. Note that there is not always an exact match between the phrase associated in the KBGen lexicon with a type and the phrase occurring in the training sentence. To account for this, we use some additional similarity based heuristics to identify the phrase in the input string that is most likely to be associated with a variable lacking an exact match in the input string. E.g., for entity variables (e.g., `Particle-In-Motion64582`), we search the input string for nouns (e.g., particles) whose overlap with the variable type (e.g., Particle-In-Motion) is not empty.

4.2.2 Inducing a based FB-LTAG from the aligned data

To extract a Feature-Based Lexicalised Tree Adjoining Grammar (FB-LTAG) from the KBGen data, we parse the sentences of the training corpus; project the entity and event variables to the syntactic projection of the strings they are aligned with; and extract the elementary trees of the resulting FB-LTAG from the parse tree using semantic information. Figure 4 shows the trees extracted from the scenario given in Figure 1.

To associate each training example sentence with a syntactic parse, we use the Stanford parser. After alignment, the entity and event variables occurring in the input semantics are associated with substrings of the yield of the syntactic parse tree. We project these variables up the syntactic tree to reflect headedness. A variable aligned with a noun is projected to the NP level or to the immediately dominating PP if it occurs in the subtree dominated by the leftmost daughter of that PP. A variable aligned with a verb is projected to the first S node immediately dominating that verb or, in the case of a predicative sentence, to the root of that sentence[3].

Once entity and event variables have been projected up the parse trees, we extract elementary FB-LTAG trees and their semantics from the input scenario as follows.

First, the subtrees whose root node is indexed with an entity variable are extracted. This results in a set of NP and PP trees anchored with entity names and associated with the predication true of the indexing variable.

Second, the subtrees capturing relations between variables are extracted. To perform this extraction, each input variable X is associated with a set of dependent variables i.e., the set of variables Y such that X is related to Y $(R(X, Y))$. The minimal tree containing all and only the dependent variables $D(X)$ of a variable X is then extracted and associated with the set of literals Φ such that $\Phi = \{R(Y, Z) \mid (Y = X \wedge Z \in D(X)) \vee (Y, Z \in D(X))\}$. This procedure extracts the subtrees relating the argument variables of a semantic functors such as an event or a role e.g., a tree describing a verb and its arguments as shown in the top

[3]Initially, we used the head information provided by the Stanford parser. In practice however, we found that the heuristics we defined to project semantic variables to the corresponding syntactic projection were more accurate and better supported our grammar extraction process.

`Particle-In-Motion`	molecule in motion,molecules in motion
`Endoplasmic-Reticulum`	endoplasmic reticulum,endoplasmic reticulum
`Gated-Channel`	gated Channel,gated Channels
`Release-Of-Calcium`	releases,release,released,release

The function of a (gated channel, `Gated-Channel64605`) is to (release, `Release-Of-Calcium646`) (particles, `Particle-In-Motion64582`) from the (endoplasmic reticulum, `Endoplasmic-Reticulum64603`)

Figure 3: Example Entries from the KBGen Lexicon and example alignment

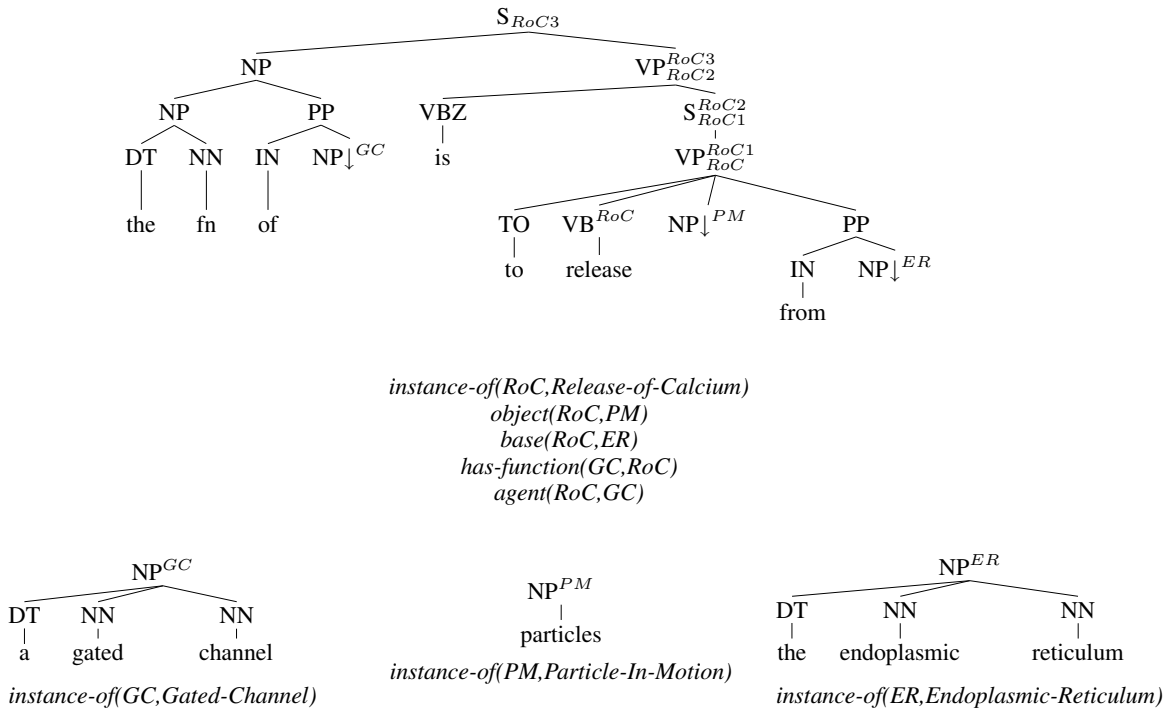

Figure 4: Extracted Grammar for "*The function of a gated channel is to release particles from the endoplasmic reticulum*". Variable names have been abbreviated and the KBGen tuple notation converted to terms so as to fit the input format expected by our surface realiser.

part of Figure 4. Note that such a tree may capture a verb occurring in a relative or a subordinate clause (together with its arguments) thus allowing for complex sentences including a relative or relating a main and a subordinate clause.

The resulting grammar extracted from the parse trees (cf. e.g., Figure 4) is a Feature-Based Tree Adjoining Grammar with a Unification-based compositional semantics as described in (Gardent and Kallmeyer, 2003). In particular, our grammars differs from the traditional probabilistic Tree Adjoining Grammar extracted as described in e.g., (Chiang, 2000) in that they encode both syntax and semantics rather than just syntax. They also differ from the semantic FB-TAG extracted by (DeVault et al., 2008) in that (i) they encode the linking between syntactic and semantic arguments; (ii) they allow for elementary trees spanning discontiguous strings (e.g., *The function of X is to release Y*); and (iii) they enforce the semantic principle underlying TAG namely that an elementary tree containing a syntactic functor also contains its syntactic arguments.

4.3 Generation

To generate with the grammar extracted from the KBGen data, we use the GenI surface realiser (Gardent et al., 2007). Briefly, given an input semantics and a FB-LTAG with a unification based semantics, GenI selects all grammar entries whose semantics subsumes the input semantics; combines these entries using the FB-LTAG combination operations (i.e., adjunction and substitution); and outputs the yield of all derived trees which are syntactically complete and whose semantics is the input semantics. To rank the generator output, we train a language model on the GeniA corpus [4], a corpus of 2000 MEDLINE asbtracts about biology containing more than 400000 words (Kim et al., 2003) and use this model to rank the generated sentences by decreasing probability.

Thus for instance, given the input semantics shown in Figure 1 and the grammar depicted in Figure 4, the surface realiser will select all of these trees; combine them using FB-LTAG substitution operation; and output as generated sentence the yield of the resulting derived tree namely the sentence *The function of a gated channel is to release particles from the endoplasmic reticulum*.

However, this procedure only works if the entries necessary to generate from the given input are present in the grammar. To handle new, unseen input, we proceed in two ways. First, we try to guess a grammar entry from the shape of the input and the existing grammar. Second, we expand the grammar by decomposing the extracted trees into simpler ones.

4.4 Guessing new grammar entries.

Given the limited size of the training data, it is often the case that input from the test data will have no matching grammar unit. To handle such previously unseen input, we start by partitioning the input semantics into sub-semantics corresponding to events, entities and role.

For each entity variable X of type $Type$, we create a default NP tree whose semantics is a literal of the form *instance-of(X,Type)*.

For event variables, we search the lexicon for an entry with a matching or similar semantics i.e., an entry with the same number and same type of literals (literals with same arity and with identical relations). When one is found, a grammar entry is constructed for the unseen event variable by substituting the event type of the matching entry with the type of the event variable. For instance, given the input semantics *instance-of(C,Carry), object(C,X), base(C,Y), has-function(Z,C), agent(C,Z)*, this procedure will create a grammar entry identical to that shown at the top of Figure 4 except that the event type *Release-of-Calcium* is changed to *Carry* and the terminal *release* to the word form associated in the KBGen lexicon with this concept, namely to the verb *carry*.

4.5 Expanding the Grammar

While the extracted grammar nicely captures predicate/argument dependencies, it is very specific to the items seen in the training data. To reduce overfitting, we generalise the extracted grammar by extracting from each event tree, subtrees that capture structures with fewer arguments and optional modifiers.

For each event tree τ extracted from the training data which contains a subject-verb-object subtree τ', we add τ' to the grammar and associate it with the semantics of τ minus the relations associated with the arguments that have been removed. For instance, given the extracted tree for the sentence *"Aquaporin facilitates the movement of water molecules through hydrophilic channels."*, this

[4] http://www.nactem.ac.uk/genia/

procedure will construct a new grammar tree corresponding to the subphrase *"Aquaporin facilitates the movement of water molecules"*.

We also construct both simpler event trees and optional modifiers trees by extracting from event trees, PP trees which are associated with a relational semantics. For instance, given the tree shown in Figure 4, the PP tree associated with the relation *base(RoC,ET)* is removed thus creating two new trees as illustrated in Figure 5: an S tree corresponding to the sentence *The function of a gated channel is to release particles* and an auxiliary PP tree corresponding to the phrase *from the endoplasmic reticulum*. Similarly in the above example, a PP tree corresponding to the phrase *"through hydrophilic channels."* will be extracted.

As with the base grammar, missing grammar entries are guessed from the expanded grammar. However we do this only in cases where a correct grammar entry cannot be guessed from the base grammar.

5 Experimental Setup

We evaluate our approach on the KBGen data and compare it with the KBGen reference and two other systems having taken part to the KBGen challenge.

5.1 Training and test data.

Following a practice introduced by (Angeli et al., 2010), we use the term *scenario* to denote a KB subset paired with a sentence. The KBGen benchmark contains 207 scenarii for training and 72 for testing. Each KB subset consists of a set of triples and each scenario contains on average 16 triples and 17 words.

5.2 Systems

We evaluate three configurations of our approach on the KBGen test data: one without grammar expansion (BASE); a second with a manual grammar expansion MANEXP; and a third one with automated grammar expansion AUTEXP. We compare the results obtained with those obtained by two other systems participating in the KBGen challenge, namely the UDEL system, a symbolic rule based system developed by a group of students at the University of Delaware; and the IMS system, a statistical system using a probabilistic grammar induced from the training data.

5.3 Metrics.

We evaluate system output automatically, using the BLEU-4 modified precision score (Papineni et al., 2002) with the human written sentences as reference. We also report results from a human based evaluation. In this evaluation, participants were asked to rate sentences along three dimensions: **fluency** (Is the text easy to read?), **grammaticality** and meaning similarity or **adequacy** (Does the meaning conveyed by the generated sentence correspond to the meaning conveyed by the reference sentence?). The evaluation was done on line using the LG-Eval toolkit (Kow and Belz, 2012), subjects used a sliding scale from -50 to +50 and a Latin Square Experimental Design was used to ensure that each evaluator sees the same number of outputs from each system and for each test set item. 12 subjects participated in the evaluation and 3 judgments were collected for each output.

6 Results and Discussion

System	All	Covered	Coverage	# Trees
IMS	0.12	0.12	100%	
UDEL	0.32	0.32	100%	
Base	0.04	0.39	30.5%	371
ManExp	0.28	0.34	83 %	412
AutExp	0.29	0.29	100%	477

Figure 6: BLEU scores and Grammar Size (Number of Elementary TAG trees

Table 6 summarises the results of the automatic evaluation and shows the size (number of elementary TAG trees) of the grammars extracted from the KBGen data.

The average BLEU score is given with respect to all input (All) and to those inputs for which the systems generate at least one sentence (Covered). While both the IMS and the UDEL system have full coverage, our BASE system strongly undergenerates failing to account for 69.5% of the test data. However, because the extracted grammar is linguistically principled and relatively compact, it is possible to manually edit it. Indeed, the MANEXP results show that, by adding 41 trees to the grammar, coverage can be increased by 52.5 points reaching a coverage of 83%. Finally, the AUTEXP results demonstrate that the automated expansion mechanism permits achieving full coverage while keeping a relative small grammar (477 trees).

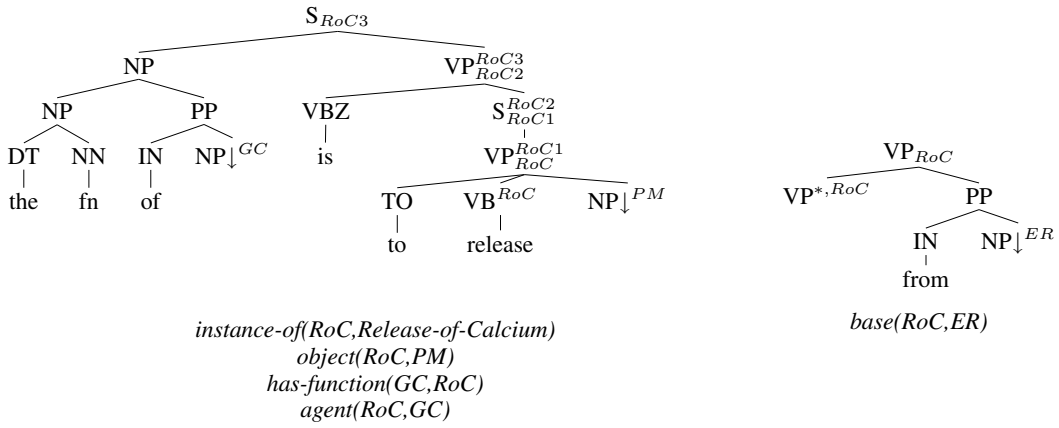

Figure 5: Trees Added by the Expansion Process

	Fluency		Grammaticality		Meaning Similarity	
System	Mean	Homogeneous Subsets	Mean	Homogeneous Subsets	Mean	Homogeneous Subsets
UDEL	4.36	A	4.48	A	3.69	A
AutExp	3.45	B	3.55	B	3.65	A
IMS	1.91	C	2.05	C	1.31	B

Figure 7: Human Evaluation Results on a scale of 0 to 5. Homogeneous subsets are determined using Tukey's Post Hoc Test with $p < 0.05$

In terms of BLEU score, the best version of our system (AUTEXP) outperforms the probabilistic approach of IMS by a large margin (+0.17) and produces results similar to the fully handcrafted UDEL system (-0.03).

In sum, our approach permits obtaining BLEU scores and a coverage which are similar to that obtained by a hand crafted system and outperforms a probabilistic approach. One key feature of our approach is that the grammar extracted from the training data is linguistically principled in that it obeys the extended locality principle of Tree Adjoining Grammars. As a result, the extracted grammar is compact and can be manually modified to fit the need of an application as shown by the good results obtained when using the MANEXP configuration.

We now turn to the results of the human evaluation. Table 7 summarises the results whereby systems are grouped by letters when there is no significant difference between them (significance level: $p < 0.05$). We used ANOVAs and post-hoc Tukey tests to test for significance. The differences between systems are statistically significant throughout except for meaning similarity (adequacy) where UDEL and our system are on the same level. Across the metrics, our system consistently ranks second behind the symbolic, UDEL system and before the statistical IMS one thus confirming the ranking based on BLEU.

7 Conclusion

In Tree Adjoining Grammar, the *extended domain of locality principle* ensures that TAG trees group together in a single structure a syntactic predicate and its arguments. Moreover, the *semantic principle* requires that each elementary tree captures a single semantic unit. Together these two principles ensure that TAG elementary trees capture basic semantic units and their dependencies. In this paper, we presented a grammar extraction approach which ensures that extracted grammars comply with these two basic TAG principles. Using the KBGen benchmark, we then showed that the resulting induced FB-LTAG compares favorably with competing symbolic and statistical approaches when used to generate from knowledge base data.

In the current version of the generator, the output is ranked using a simple language model trained on the GENIA corpus. We observed that this often fails to return the best output in terms of BLEU score, fluency, grammaticality and/or meaning. In the future, we plan to remedy this using a ranking approach such as proposed in (Velldal and Oepen, 2006; White and Rajkumar, 2009).

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, Donia Scott, Nikhil Dinesh, and Fennie Liang. 2012. Kbgen: Text generation from knowledge bases as a new shared task. In *Proceedings of the seventh International Natural Language Generation Conference*, pages 141–145. Association for Computational Linguistics.

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

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.

David Chiang. 2000. Statistical parsing with an automatically-extracted tree adjoining grammar. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 456–463. Association for Computational Linguistics.

Peter Clark and Bruce Porter. 1997. Building concept representations from reusable components. In *AAAI/IAAI*, pages 369–376. Citeseer.

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.

David DeVault, David Traum, and Ron Artstein. 2008. Making grammar-based generation easier to deploy in dialogue systems. In *Proceedings of the 9th SIGdial Workshop on Discourse and Dialogue*, pages 198–207. Association for Computational Linguistics.

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.

Claire Gardent and Laura Kallmeyer. 2003. Semantic construction in feature-based tag. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-Volume 1*, pages 123–130. Association for Computational Linguistics.

Claire Gardent, Eric Kow, et al. 2007. A symbolic approach to near-deterministic surface realisation using tree adjoining grammar. In *ACL*, volume 7, pages 328–335.

B. Grosof. 2012. The silk project: Semantic inferencing on large knowledge. Technical report, SRI. `http://silk.semwebcentral.org/`.

D. Gunning, V. K. Chaudhri, P. Clark, K. Barker, Shaw-Yi Chaw, M. Greaves, B. Grosof, A. Leung, D. McDonald, S. Mishra, J. Pacheco, B. Porter, A. Spaulding, D. Tecuci, and J. Tien. 2010. Project halo update - progress toward digital aristotle. *AI Magazine*, Fall:33–58.

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

Martin Kay. 1996. Chart generation. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, pages 200–204. Association for Computational Linguistics.

Joohyun Kim and Raymond J Mooney. 2010. Generative alignment and semantic parsing for learning from ambiguous supervision. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 543–551. Association for Computational Linguistics.

J-D Kim, Tomoko Ohta, Yuka Tateisi, and Junichi Tsujii. 2003. Genia corpusa semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl 1):i180–i182.

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.

Eric Kow and Anja Belz. 2012. Lg-eval: A toolkit for creating online language evaluation experiments. In *LREC*, pages 4033–4037.

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.

Wei Lu and Hwee Tou Ng. 2011. A probabilistic forest-to-string model for language generation from typed lambda calculus expressions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1611–1622. Association for Computational Linguistics.

Wei Lu, Hwee Tou Ng, and Wee Sun Lee. 2009. Natural language generation with tree conditional random fields. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 400–409. Association for Computational Linguistics.

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*.

Boris Motik, Peter F Patel-Schneider, Bijan Parsia, Conrad Bock, Achille Fokoue, Peter Haase, Rinke Hoekstra, Ian Horrocks, Alan Ruttenberg, Uli Sattler, et al. 2009. Owl 2 web ontology language: Structural specification and functional-style syntax. *W3C recommendation*, 27:17.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

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

R. Power and A. Third. 2010. Expressing owl axioms by english sentences: dubious in theory, feasible in practice. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 1006–1013. Association for Computational Linguistics.

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

Hadar Shemtov. 1996. Generation of paraphrases from ambiguous logical forms. In *Proceedings of the 16th conference on Computational linguistics-Volume 2*, pages 919–924. Association for Computational Linguistics.

Stuart M Shieber, Gertjan Van Noord, Fernando CN Pereira, and Robert C Moore. 1990. Semantic-head-driven generation. *Computational Linguistics*, 16(1):30–42.

Erik Velldal and Stephan Oepen. 2006. Statistical ranking in tactical generation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 517–525. Association for Computational Linguistics.

K. Vijay-Shanker and AK Joshi. 1988. Feature structures based tree adjoining grammars. In *Proceedings of the 12th International Conference on Computational Linguistics*, Budapest, Hungary.

Juen-tin Wang. 1980. On computational sentence generation from logical form. In *Proceedings of the 8th conference on Computational linguistics*, pages 405–411. Association for Computational Linguistics.

Michael White and Rajakrishnan Rajkumar. 2009. Perceptron reranking for ccg realization. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 410–419. Association for Computational Linguistics.

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.