Using C5.0 and Exhaustive Search for Boosting Frame-Semantic Parsing Accuracy

Guntis Barzdins, Didzis Gosko, Laura Rituma, Peteris Paikens
Institute of Mathematics and Computer Science, University of Latvia
Rainis Blvd 29, Riga LV-1459, Latvia
E-mail: guntis.barzdins@lumii.lv, didzis.gosko@gmail.com, {laura, peteris}@ailab.lv

Abstract

Frame-semantic parsing is a kind of automatic semantic role labeling performed according to the FrameNet paradigm. The paper reports a novel approach for boosting frame-semantic parsing accuracy through the use of the C5.0 decision tree classifier, a commercial version of the popular C4.5 decision tree classifier, and manual rule enhancement. Additionally, the possibility to replace C5.0 by an exhaustive search based algorithm (nicknamed C6.0) is described, leading to even higher frame-semantic parsing accuracy at the expense of slightly increased training time. The described approach is particularly efficient for languages with small FrameNet annotated corpora as it is for Latvian, which is used for illustration. Frame-semantic parsing accuracy achieved for Latvian through the C6.0 algorithm is on par with the state-of-the-art English frame-semantic parsers. The paper includes also a frame-semantic parsing use-case for extracting structured information from unstructured newswire texts, sometimes referred to as bridging of the semantic gap.

Keywords: FrameNet, semantic role labelling, information extraction

1. Introduction

Development of FrameNet [1] resources for various languages is an ongoing activity (Burchardt at al., 2006; Leenoi at al., 2011). Much of that effort is aimed at only mapping the English FrameNet frames into lexical and syntactic structures of other languages and thus creating a FrameNet annotated corpora for the target language. Meanwhile creation of a Latvian FrameNet was motivated primarily by computational needs of automatic information extraction from natural language texts (predominantly newswire articles). The benchmark methodology for automatic frame-semantic parsing was set at SemEval-2007 (Baker at al., 2007) and specifically - by the best performing LTH system (Johansson & Nugues, 2007). Further improvements to the methodology were implemented in the state-of-art SEMAFOR system (Das at al., 2014).

In this paper we report a novel approach for boosting frame-semantic parsing accuracy through the use of the C5.0 decision tree classifier[2] (Quinlan, 1993) and manual rule enhancement. We also describe a possibility to replace C5.0 by exhaustive search (nicknamed "C6.0") leading to even higher frame-semantic parsing accuracy. This approach is particularly efficient for languages with small FrameNet annotated corpora as is the case for Latvian, which is used in this paper for illustration.

2. Latvian FrameNet

Latvian FrameNet originally was created for a practical information extraction system (described in Section 5) developed for a national news agency to automatically extract biographical data about publicly visible persons and organizations mentioned in the newswire articles. A number of design decisions were taken to strengthen the computational nature of Latvian FrameNet.

First design decision was to preprocess all input texts with a tokenizer and POS tagger (Paikens at al., 2013), an unlabeled [3] dependency parser (Pretkalnina & Rituma, 2013; Pretkalnina et al., 2014), and a NER and co-reference resolver (Znotins & Paikens, 2014) to produce extended CoNLL-style annotations prior to any FrameNet annotation (see Fig.1).

Index	Form (Word)	Lemma	POS	Tag	Parent	Named Entity Type (NER)	Named Entity ID
1	pienākumus	pienākums	n	ncmpa1	3	O	
2	sāks	sākt	v	vmnif130an	3	O	
3	pildīt	pildīt	v	vmnn0t3000n	0	O	
4	pašreizējais	pašreizējs	a	armsnyp	6	O	185
5	Latvijas	Latvija	n	npfsg4	6	location	182
6	vēstnieks	vēstnieks	n	ncmsn1	3	profession	183
7	ASV	ASV	y	y	6	profession	184
8	Ojārs	ojārs	n	n_msn1	9	person	183
9	Kalniņš	kalniņš	n	ncmsn1	7	person	183
10	,	,	z	zc	3	O	

Figure 1: CoNLL style input data for FrameNet tools, a sentence „*Duties began performing current Latvia ambassador to USA Ojars Kalnins."* preprocessed with POS, unlabeled dependency, NER, co-reference parsers

Secondly, a novel FrameNet graphical editor[4] (Fig. 2) was developed (Brediks, 2013) specifically for annotating dependency pre-parsed texts illustrated in Fig 1. The key difference from the legacy phrase-structure grammar based Berkeley FrameNet annotation tool (Ruppenhofer at al., 2010) or the Salto FrameNet annotation tool (Burchardt at al., 2006) is that our tool relies on the dependency-tree to automatically derive filler phrase boundaries once the head-word for the frame element (FE) is selected. This tool was used to create a FrameNet annotated corpus for Latvian. The corpus currently

[1] http://framenet.icsi.berkeley.edu

[2] C5.0 is a commercial version of C4.5 – a decision tree classifier popular for data mining applications, available from http://rulequest.com/see5-info.html

[3] Labeled dependency trees are used in Section 4 to improve the handling of coordination

[4] http://www.ltn.lv/~guntis/FrameMarker.zip

contains almost 5000 sentences from various types of newswire sources.

Third design decision was to use a reduced number of frames – although our methodology is applicable to any number of frames, we have selected just 26 Frames (*Being born, People by age, Death, Personal relationship, Being named, Residence, Education teaching, People by vocation, People by origin, Being employed, Hiring, Employment end, Membership, Change of leadership, Giving, Intentionally create, Participation, Earnings and losses, Public procurement, Possession, Lending, Trial, Attack, Win prize, Statement, Product line*) which were of interest to the national news agency for media monitoring purposes; this use-case dictated also adding or removal of some frame elements (arguments) as shown in Fig. 3.

3. Frame-Semantic Parsing

Thanks to above design decisions it was rather straightforward to adapt the benchmark LTH frame-semantic parser (Johansson & Nugues, 2007) approach to Latvian FrameNet. The original LTH frame-semantic parser uses multiple SVM classifiers to identify frame targets and frame elements. Besides SVM we explored various machine learning approaches, including a log-linear implementation of SEMAFOR [5] system, but the achieved accuracy turned out to be low due to limited size of available FrameNet annotated corpora for Latvian. This problem lead to the key innovation reported in this paper – the C5.0 based manual boosting of frame-semantic annotation accuracy.

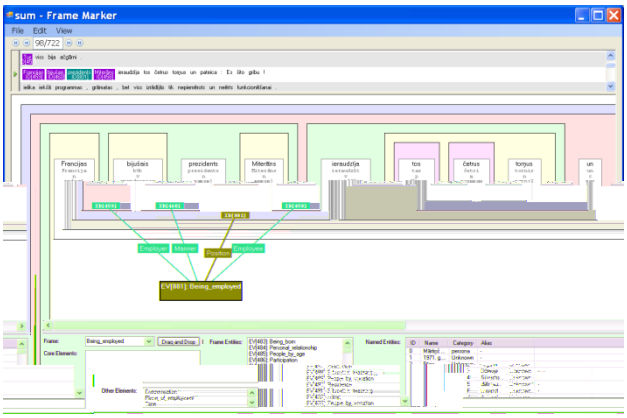

Figure 2: Dependency-tree based FrameNet editor

In terms of classification accuracy C5.0 (C4.5) is comparable to SVM (Shawkat & Smith, 2006) although C5.0 is typically used with lesser training data sets than SVM. Meanwhile the crucial advantage of C5.0 (C4.5) is that the decision tree classifier generated automatically from corpus can be output also in the form of human readable and editable rules like shown below:

```
Rule 1: (5, lift 585.8)
     PreviousLEMMA = euro    (Euro)
     CurrentLEMMA = apgrozījums    (turnover)
     -> class YES (Earnings_and_losses)  [0.857]
Rule 2: (9/1, lift 559.5)
     CurrentLEMMA = peļņa    (profit)
     NextLEMMA = būt    (be)
     -> class YES (Earnings_and_losses)  [0.818]
```

Such classification rules can be easily (effort of approximately 1 hour per frame type) enhanced manually by a human linguist to significantly boost accuracy of frame-semantic parser. Typical rule-changes made by human linguist are adding complete list of month-names, if "January" is mentioned in the rule, or adding more professions, if "plumber" appears in the rule, or discarding some silly rules caused by training data sparsity. Tables 2. and 3. show the actual boosting effect achieved. One can observe that manual boosting results in increased precision (at the expense of slightly reduced recall in case of frame element recognition). It is crucial to note that such manual boosting is quite "cheap" compared to effort required to achieve a similar boost by merely annotating more training data. To achieve simpler classification rules to be read and edited by human, we trained a separate[6] binary (YES/NO) C5.0 classifier for identification of each frame target and frame element type. This is slightly different from the approach taken in LTH frame-semantic parser, which divides the task into the following steps:

1) Identifying the words that should be associated with frames
2) Classifying the frames associated with the word in (1)
3) Identifying the words that should be associated with frame elements (arguments)
4) Classifying the frame elements associated with the words in (3)

In our frame-semantic parser "frame target identification" refers to steps (1) and (2) jointly, as these are handled by one binary C5.0 classifier per frame type, which merely classifies if the current word in the text is (or is not) a target for this specific frame type. Similarly "frame element identification" in our case refers to (3) and (4) jointly and is handled by one binary C5.0 classifier per frame element type.

In our approach positive examples in the resulting training datasets are sparse and require considerable tweaking of C5.0 parameters to produce meaningful rules. We concluded on the following command-line parameter settings:

```
$ ./c5.0 -r -m1 -c100 -f <name>
```

along with associated <name>.costs file heavily penalizing missed YES-rules: "`NO, YES: 100`"

[5] Log-linear or perceptron based approaches have significant drawback (compared to kernelized SVM or C5.0) – besides the list of basic features they require also "feature templates" to handle feature vector value patterns. These feature patterns need to be manually crafted by the domain expert (Das at al., 2014). Use of C4.5 to automate feature template generation (Fernandes & Milidi'u, 2012) was seminal to the approach described in this paper.

[6] Our approach of identifying each frame type separately allows to scale it linearly from 26 frames in Latvian FrameNet to over 1000 frames in the current English FrameNet 1.5

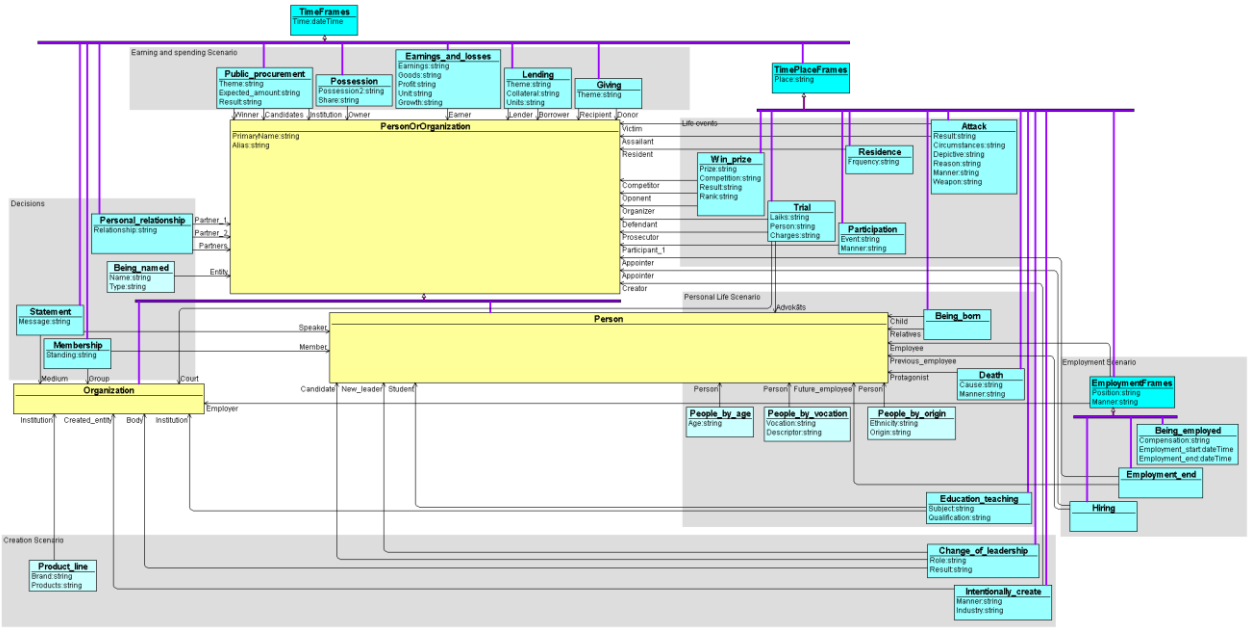

Figure 3: Latvian FrameNet 26 Frames (blue boxes) and frame element filler types / NER categories (yellow).

For fully automatic frame target identification mode rules were cut-off at Laplace ratio 0.1 to avoid target overgeneration due to manipulated costs file. Frame targets are identified first and then frame element candidates are considered only in the radius of 4 words around the identified frame target word according to the dependency tree; only one frame element of a kind is retained for each frame target if C5.0 classifier found multiple candidates (the closest ones to the target according to the dependency tree).

	Latvian FrameNet data	*English SemEval '07 data*
Exemplar sentences	1682	139439
Frame labels (Frame types)	26	665
Role labels (FE types)	80	720
Sentences in test data	199	120

Table 1: FrameNet data sets used for evaluation

Target identification	*Precision*	*Recall*	*F1*
C5.0 fully automatic (Latvian FrameNet data)	51.7	39.5	44.8
C5.0 manual boosting (Latvian FrameNet data)	**55.6**	**43.9**	**49.1**
C6.0 fully automatic (Latvian FrameNet data)	**62.5**	**46.8**	**53.5**
LTH (English SemEval'07 data)	66.2	50.6	57.3
SEMAFOR (English SemEval'07 data)	69.7	54.9	61.4

Table 2: Frame target recognition evaluation results

FE identification	*Precision*	*Recall*	*F1*
C5.0 fully automatic (Latvian FrameNet data)	54.6	43.8	48.6
C5.0 manual boosting (Latvian FrameNet data)	**59.4**	**43.3**	**50.1**
C6.0 fully automatic (Latvian FrameNet data)	**61.3**	**60.7**	**61.0**
LTH (English SemEval'07 data)	51.6	35.4	42.0
SEMAFOR (English SemEval'07 data)	58.1	38.8	46.5

Table 3: Frame element recognition evaluation results

Evaluation of our initial results shows that C5.0 decision tree based approach provides accuracy that is competitive for Frame-semantic parsing, and can also be conveniently combined with manually enhanced rules for accuracy boosting. Comparing Latvian frame-semantic parsing results to state-of-the-art English frame-semantic parser accuracies suggests[7] that C5.0 and smaller size of Latvian FrameNet contributes positively to frame element recognition accuracy, while for frame target recognition corpus size is crucial. It shall be noted that for target identification English frame-semantic parsers actually use two additional information sources not available for Latvian – the list of lexical units known to invoke particular frame (lexical units are part of English FrameNet distribution) and WordNet synsets (Fellbaum, 1998).

4. Exhaustive Search (C6.0)

While experimenting with C5.0 as described in the previous Section, we noted that use of approximate

[7] Evaluation results for English were copied from [4]. Evaluation script used for English is not available online. Our evaluation script counts only exact head-word matches for frame targets and for frame elements in correctly identified frames

entropy-based C5.0 is somewhat obsolete for tasks requiring only binary classifier (e.g. our frame-semantic parser implementation), because the number of hypothetical rules recognizing positive exemplars is merely *number-of-positive-exemplars* × $2^{feature\text{-}count}$, which is a tractable number for exhaustive search up to approximately 20 features (we use 11 features for frame target identification and 13 features for frame element identification). It shall be noted that exhaustive search applies only to the rule learning stage – the runtime application of the learned rules is very fast.

Additional motivation to replace C5.0 was the costs file, which had to be manually tweaked to generate rules from unbalanced training data containing massive amounts of negative exemplars and very sparse positive exemplars. Without costs file C5.0 often gave just single default rule "negative", which is true for 99.9% of training exemplars. Few optimizations allowed cutting down the computation time for exhaustive search below one minute per classifier for the amount of training data available in Latvian FrameNet. The resulting exhaustive search based classifier we nicknamed[8] in this paper "C6.0" since for frame-semantic parsing applications it clearly surpasses the original C5.0 (including also the manually boosted C5.0 rules) – see the initial C6.0 results in Tables 2. and 3. Attempts to further manually boost the rules generated by C6.0 were nearly fruitless and improved accuracy by statistically insignificant values of less than 1%.

```
1 [_, _, _, {peļņa, apgrozījums}, _, _, _, _, _, _, _]     136  31
2 [_, ng, _, zaudējums, _, _, _, _, _, _, _]               10   0
3 [_, _, _, {zaudējums, ienākums}, _, nn, _, _, _, _, _]   12   2
4 [_, _, _, nopelnīt, _, _, _, _, _, x, _]                 6    0
5 [uzņēmums, _, _, _, _, _, _, _, _, vcnpa, _]             2    0
6 [kompānija, _, _, _, _, v_nia, _, _, _, _, _]            2    0
7 [',', _, _, ieņēmums, _, _, _, _, _, _, _]               2    0
```

Figure 4: C6.0 generated target identification rules for frame *Earnings and losses*. Shown are counts in the training corpus for total matches and false positives.

Meanwhile the human-readable, optimal rules generated by C6.0 (it is actually quite insightful to read these machine generated rules, see Fig. 4) opened two other possibilities for boosting the frame-semantic parsing accuracy:

a. Correcting the frame annotation inconsistencies in the training corpus.
b. Spotting the missing features preventing C6.0 from inferring universal rules with high coverage.

Training corpus annotation inconsistencies are particularly easy to spot in the human-readable frame target identification rules generated by C6.0, because these rules substitute for the meaningful lists of lexical units (word senses, included in the English FrameNet distribution) known to invoke the particular frame. Meanwhile frame element identification rules generated by C6.0 correspond to meaningful lexical entries [9] (containing frame element syntactic realization variations in the annotated corpora) in English FrameNet distribution. Tables 4, 5, 6 show the final results after the spotted annotation inconsistencies (mostly they were missed frames) were corrected in the extended training corpus and few missing features were added to the parser. Fig. 5 shows cross-validation of frame target F1 score relative to various split of training and test sets. The evaluation results show that the resulting C6.0 based Latvian frame-semantic parser performs on par with state-of-the-art English frame-semantic parsers despite smaller FrameNet training corpus for Latvian.

	Latvian FrameNet data	*English SemEval '07 data*
Exemplar sentences	4079	139439
Frame labels (Frame types)	26	665
Role labels (FE types)	80	720
Sentences in test data	844	120

Table 4: Extended data sets used for evaluation

Target identification	*Precision*	*Recall*	*F1*
C6.0 fully automatic (Latvian FrameNet data)	**63.5**	**62.7**	**63.1**
LTH (English SemEval'07 data)	66.2	50.6	57.3
SEMAFOR (English SemEval'07 data)	69.7	54.9	61.4

Table 5: Frame target recognition final results

FE identification	*Precision*	*Recall*	*F1*
C6.0 fully automatic (Latvian FrameNet data)	**65.9**	**76.8**	**70.9**
LTH (English SemEval'07 data)	51.6	35.4	42.0
SEMAFOR (English SemEval'07 data)	58.1	38.8	46.5

Table 6: Frame element recognition final results

The final list of features used for frame target identification was:

PLEMMA – previous word lemma
PPOS –previous word morphology tag
PNETYPE – previous word NE type
LEMMA – target word lemma
LEMMA_CLUSTER – target word cluster
POS – target word morphology tag
DEPLABEL – syntax role of the target word
NETYPE – target word NE type

[8] C6.0 is not a universal substitute for the much richer functionality of C5.0 useful in other application domains

[9] Lexical entries in English FrameNet include also valence patterns, defining meaningful frame element subsets and their syntactic realizations observed in the annotated corpora; in our parser meaningful frame element subsets are hardcoded

NLEMMA – next word lemma
NPOS – next word morphology tag
NNETYPE – next word NE type

The final list of features used for frame element identification was:

LEMMA – FE headword lemma
LEMMA_CLUSTER – FE headword lemma cluster
POS – FE headword morphology tag
NETYPE – FE headword NE type
DEPLABEL – syntax role of the FE headword
HLEMMA – parent word lemma
HLEMMA_CLUSTER – parent word cluster
HPOS – parent word morphology tag
HNETYPE – parent word NE type
TARGET_TYPE – frame name
TARGET_PATH2D – sequence of 4-direction moves forming the path in the dependency tree between FE headword and target word
TARGET_PATH2D_SHORT – the path without sequential duplicates
TARGET_NEAR – path length above or below 4

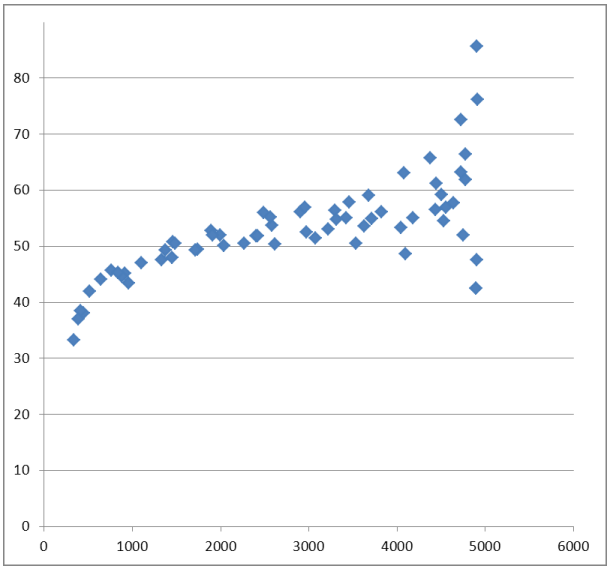

Figure 5: Dynamics of frame target F1 score relative to the number of sentences in the training set versus test set. The total number of annotated sentences is 4923.

The actual implementation of C6.0 algorithm we have developed is slightly more sophisticated than pure relaxation of positive exemplars for exhaustive search of best rules shown in Fig. 4, as algorithm has to decide which of the searched rules form the best rule-set without falling victim to the overfitting/underfitting problem. Overfitting occurs when rules have too high precision at the expense of low recall – such rules perform excellent on the training set, but are not general enough to be useful for unseen data. Underfitting is the opposite extreme, where high recall is achieved at the expense of low precision due to rules being too promiscuous. In C6.0 we use the same approach as C5.0 to address the overfitting/underfitting problem through confidence limits for the binomial distribution or through Laplace ratio. The best F1 scores we achieved with the default Laplace ratio *(n-m+1)/(n+2)* for rule's accuracy estimation, where *n* is the number of exemplars covered by the rule and *m* shows how many of them are false positives (*n* and *m* are the two numbers shown in Fig. 4 for every rule). Meanwhile confidence limits for the binomial distribution gave better recall rates with slight degradation to precision and overall F1 accuracy.

The actual C6.0 implementation [10] includes minor additional fine-tuning options such as tiebreaking strategy for rules with equal Laplace ratio – preferring the most relaxed or the most specific rule (default is choosing the most specific rule) and restricting the maximum number of features appearing in one rule (default is 5, although 3 gives nearly as good results in the fraction of time). C6.0 also includes sieves to minimize the number of overlapping rules and to keep only rules covering more than one exemplar, as fewer rules in the resulting rule-set tend to improve the overall accuracy on unseen data.

5. Discussion

The ability to achieve high accuracy for frame-semantic parsing enables streamlining of information extraction task from natural language texts, such as newswire articles. The goal of such information extraction effectively is populating the ontology[11] shown in Fig.3 (this is OWLGrEd[12] visualization of the actual OWL ontology) with instance data retrieved from the text. To do so, frame-semantic parsing techniques described in this paper (producing instances for the blue boxes in Fig.3) need to be combined with Cross Document Coreference (CDC) techniques (Wick at al., 2013) to automatically determine which mentions in the text refer to the same real-world entity (producing disambiguated instances for the yellow boxes in Fig.3).

We have implemented such integrated information extraction system and populated it with data from approximately 1 million newswire articles. From the practical standpoint it turned out that the bottleneck of the approach is Named Entity discovery and linking accuracy – even at estimated 80% CDC accuracy it too often merged together different real-world entities with similar names or did not link together alternative spellings for the same entity (due to frame elements often being a hierarchy of Named Entities, e.g. "*triju Zvaigžņu ordeņa virsnieks*" in Fig. 6), making the overall results unusable. To mitigate the problem, we deflected to the use of the predefined Knowledge Base of manually disambiguated well-known person, organization, location, product, event names (with their commonly used aliases), which can be identified in the text more robustly using Named Entity linking methods similar to DBpedia Spotlight (Daiber at al., 2013). Of course, this workaround links only frame elements found in the predefined Knowledge Base, leaving other frame element fillers unidentified. The

[10] http://c60.ailab.lv
[11] http://www.ltn.lv/~guntis/FrameNetLV.owl
[12] http://owlgred.lumii.lv

unidentified frame element fillers therefore are stored as simple text strings as they appear in the original sentences (technically they can be stored in the same Knowledge Base, only tagged as "unidentified entities").
From the practical standpoint of information extraction about persons and organizations from the newswire texts this has turned out to be the best solution – link only entities present in the Knowledge Base, but leave all other frame element fillers identified only by the text strings as they appear in the source text. This mixed approach allows for creating a convenient user interface, where instance data from the Knowledge Base in Fig. 3 is verbalized using a light version of (Dannells & Gruzitis, 2014) producing simple sentences as illustrated in Fig. 6 which can further be formatted in the familiar Curriculum Vitae like manner.

Ieva Akuratere bija solista amatā [23]
Ieva Akuratere bija Puķu burves amatā [8]
Ieva Akuratere bija mūziķes un aktrises amatā [5]
Ieva Akuratere bija deputātes amatā Rīgas domē [4]
Ieva Akuratere bija solista amatā Koncertuzvedumā [4]
Ieva Akuratere bija dziedātājas amatā [3]
Ieva Akuratere bija triju Zvaigžņu ordeņa virsnieka amatā Latvijā [3]

Figure 6: Fragment of the automatically generated person profile (verbalization of *Being employed* frame). Linked Named Entities underlined, duplicate counts in brackets.

Although not yet implemented in a practical system, there is a further refinement possible for the above described Knowledge Base and information extraction system – adding the time dimension (in Fig. 3 note that *Time* is the dominant frame element present in almost all frames). For most frames extracted from the newswire texts the time of their occurrence is either explicitly specified in the text and can be retrieved by frame-semantic parser as frame element *Time* or approximate time can be retrieved from the metadata of the newswire article publication date.
Having time associated with all extracted frames opens a possibility (Barzdins, 2011) for structuring the information extracted from the newswire texts – rather than having a mix of seemingly contradictory facts in one Knowledge Base (e.g. *"Peter lives in Paris"* and *"Peter lives in NewYork"*) we can create a whole sequence of Knowledge Base instances (one per every day of history), with each instance containing only the facts which were true on that particular day and thus make these instances non-contradictory (e.g. *"Peter lives in Paris" (in instances for 2001)* and *"Peter lives in New York" (in in instances for 2011)*). Inserting frames extracted from the text by the frame-semantic parser into the proper instance (or sequence of instances) of the Knowledge Base is not an easy task (Murray & Singliar, 2012), as some frames describe an instantaneous event (e.g. frame *Attack*) while other frames describe a state which is true over prolonged period of time (e.g. frame *Being employed*). Nevertheless, resolving the time dimension (and for some sorts of tasks – also spatial dimension) is a vital additional tool for truly bridging the semantic gap in natural language understanding, eventually enabled by the accurate frame-semantic parsing.

Being born	100	Residence	67	Participation	40
Earnings and losses	89	Statement	67	Employment end	33
Death	80	Hiring	62	Product line	33
Education teaching	71	Membership	50	Lending	29
Being employed	70	Possession	48	Personal relationship	25
Change of leadership	67	People by vocation	46	Trial	18
Intentionally create	67	Win prize	45	People by origin	16

Table 7: Target identification F1 scores for some Latvian FrameNet frames.

To evaluate to what extent the information extraction approach described in this paper actually bridges the semantic gap (Ehrig, 2007) between the unstructured newswire input text and the structured output (Knowledge Base or ontology in Fig. 3), Table 7 breaks down the target identification accuracy for various frames. These results illustrate that target identification accuracy varies widely between different frame types, meaning that the current set of features apparently is not sufficient for identification of the low-scoring frames. Another explanation for the low-scoring frames might be that the concept they convey is broader (can be expressed in more ways) and thus bridging of the semantic gap with high accuracy for these frames requires a larger training corpus.

6. Conclusion

The described approach illustrates the possibility of bootstrapping a state-of-the-art frame-semantic parser for a new language by merely hand-annotating approximately 5000 sentences with the frames of interest. In our approach each frame is learned independently, meaning that the result holds for any number of different frames. It is interesting to observe that rules for frame target and frame element identification generated automatically by C6.0 effectively substitute for the manually crafted lexical unit entries which are part of the English FrameNet distribution.
On a more philosophical level, we believe that our C5.0/C6.0 based approach of statistical learning of human readable (and human-editable) rules from a corpus bridges the gap between statistical and rule-based NLP approaches and likely can be extended to other NLP areas such as the MaltParser shift-reduce dependency parsing algorithm (Nivre, et al., 2007), where the SVM classifier could be replaced by C5.0 or C6.0 to achieve a similar manual accuracy boosting effect.
Another notable achievement of C6.0 is practical machine learning by exhaustive search, which is shown to achieve high accuracy even from a small set of exemplars as shown in Fig. 5. We suspect that C6.0 is more accurate than approximate machine learning techniques popular today, but a thorough comparison with other machine learning approaches is beyond the scope of this paper.

7. Acknowledgement

The research was partially supported by Latvian 2010.-2014. National Research Program Nr.2 "Development of Innovative Multifunctional Materials, Signal Processing and Information Technologies for Competitive Science Intensive Products", project Nr.5. We also thank the anonymous reviewers for their improvement recommendations.

8. References

Baker, C., Ellsworth, M., Erk, K. (2007). SemEval-2007 task 19: Frame semantic structure extraction. In *Proceedings of SemEval-2007: 4th International Workshop on Semantic Evaluations*. Prague, pp. 99–104.

Barzdins, G. (2011). When FrameNet meets a Controlled Natural Language. In *Proceedings of NODALIDA*. Riga, NEALT Proceedings Series Vol. 11, pp. 2--5.

Brediks, D. (2013). FrameNet Semantic Annotation Editor with Co-reference Identification, Postgraduate Thesis in Informatics, University of Latvia.

Burchardt, A., Erk, K., Frank, A., Kowalski, A., Pado, S., Pinkal, M. (2006). The SALSA corpus: a German corpus resource for lexical semantics. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*. Genoa, Italy, p. 6.

Daiber, J., Jakob, M., Hokamp, C., Mendes, P.N. (2013). Improving efficiency and accuracy in multilingual entity extraction, In *Proceedings of the 9th International Conference on Semantic Systems*. ACM, pp. 121--124.

Dannells, D., Gruzitis, N. (2014). Extracting a bilingual semantic grammar from FrameNet-annotated corpora. In *Proceedings of the 9th Language Resources and Evaluation Conference (LREC)*. Reykjavik, *this volume.*

Das, D., Chen, D., Martins, A.F.T, Schneider, N., Smith, N.A. (2014). Frame-Semantic Parsing, *Computational Linguistics*, 40(1), pp. 9--56.

Ehrig, M. (2007). Ontology Alignment – Bridging the Semantic Gap. *Semantic Web and Beyond*, Vol. 4, Springer.

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

Fernandes, E.R., Milidi´u, R.L. (2012). Entropy guided feature generation for structured learning of Portuguese dependency parsing. In *Proceedings of the Conference on Computational Processing of the Portuguese Language (PROPOR)*. Lecture Notes in Computer Science, Vol. 7243, pp. 146--156.

Johansson, R., Nugues, P. (2007). LTH: semantic structure extraction using nonprojective dependency trees. In *Proceedings of SemEval-2007: 4th International Workshop on Semantic Evaluations*. Prague, pp. 227--230.

Leenoi, D., Jumpathong, S., Porkaew, P., Supnithi, T. (2011). Thai FrameNet Construction and Tools, *International Journal on Asian Language Processing*, 21(2), pp. 71--82.

Murray, W., Singliar, T. (2012). Spatiotemporal Extensions to a Controlled Natural Language. In Proceedings of the 3rd Workshop on Controlled Natural Language, volume 7427 of LNCS, pp. 61-78.

Nivre, J., Hall, J., Nilsson, J., Chanev, A., Eryigit, G., Kubler, S., Marinov, S., Marsi, E. (2007). MaltParser: A languageindependent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2):95–135.

Paikens, P., Rituma, L., Pretkalniņa, L. (2013). Morphological analysis with limited resources: Latvian example. In *Proceedings of NODALIDA*. Oslo, pp. 267--278.

Pretkalnina, L., Znotins, A., Rituma, L., Gosko, D. (2014). Dependency parsing representation effects on the accuracy of semantic applications — an example of an inflective language. In *Proceedings of the 9th Language Resources and Evaluation Conference (LREC)*. Reykjavik, *this volume.*

Pretkalnina, L., Rituma, L. (2013). Statistical syntactic parsing for Latvian. In *Proceedings of NODALIDA*. Oslo, pp. 279—290.

Quinlan, J.R. (1993). *C4.5: Programs for Machine Learning.* Morgan Kaufmann Publishers.

Ruppenhofer, J., Ellsworth, M., Petruck, M.R.L., Johnson, C.R., Scheffczyk, J. (2010). *FrameNet II: Extended Theory and Practice.* Berkeley, CA, USA: International Computer Science Institute.

Shawkat, A., Smith, K.A. (2006). On learning algorithm selection for classification. *Applied Soft Computing*, 6(2), pp. 119--138.

Wick, M., Singh, S., Pandya, H., McCallum, A. (2013). A Joint Model for Discovering and Linking Entities, In *Proceedings of the 2013 workshop on Automated knowledge base construction*. ACM, pp. 67--72.

Znotins, A., Paikens, P. (2014). Coreference Resolution for Latvian. In *Proceedings of the 9th Language Resources and Evaluation Conference (LREC)*. Reykjavik, *this volume.*