

HFST-SweNER – A New NER Resource for Swedish

Dimitrios Kokkinakis[§], Jyrki Niemi[±], Sam Hardwick[±],
Kristen Lindén[±], Lars Borin[§]

[§]Språkbanken, Department of Swedish
University of Gothenburg, Sweden
E-mail: first.last@svenska.gu.se

[±]Department of Modern Languages
University of Helsinki, Finland
E-mail: first.last@helsinki.fi

Abstract

Named entity recognition (NER) is a knowledge-intensive information extraction task that is used for recognizing textual mentions of entities that belong to a predefined set of categories, such as locations, organizations and time expressions. NER is a challenging, difficult, yet essential preprocessing technology for many natural language processing applications, and particularly crucial for language understanding. NER has been actively explored in academia and in industry especially during the last years due to the advent of social media data. This paper describes the conversion, modeling and adaptation of a Swedish NER system from a hybrid environment, with integrated functionality from various processing components, to the Helsinki Finite-State Transducer Technology (HFST) platform. This new HFST-based NER (HFST-SweNER) is a full-fledged open source implementation that supports a variety of generic named entity types and consists of multiple, reusable resource layers, e.g., various n-gram-based named entity lists (gazetteers).

Keywords: named entity recognition, finite-state transducers, Swedish

1. Introduction

Named entity recognition (NER) is a fundamental natural language processing (NLP) technology that aims at the automatic resolution of lexical ambiguity on the level of so called named entities (Sekine and Ranchhod, 2009). This is a task that is performed by locating, extracting and classifying named entities into predefined target named entity classes in unstructured text. In many cases, such named entities can be specific to a particular goal and domain (Ananiadou et al., 2004; Dozier et al., 2009; Wang 2009), but in the general case, NER systems are concerned with the identification and classification of names into a set of predefined generic classes. There are three universally accepted categories of such classes recognized, namely *persons*, *organizations* and *locations*. This set is usually used and supplemented with several other entity types, at least, such as *date/time expressions* and *numerical expressions* (e.g. *quantities*, *monetary values*, *percentages*) as well as domain-specific entities (e.g. *bibliographic references*, *names of diseases* or *names of transportation means*).

Most importantly, NER is, in principle, a prerequisite step for numerous advanced processing tasks, including ontology population (i.e., creating instances of particular concepts in a given ontology; Giuliano and Gliozzo, 2008), relation extraction (i.e., by creating structured knowledge bases with ontological relationships; Suchanek et al. 2006); text classification (Kumaran and Allan, 2004); question answering (i.e., many fact-based answers to questions are entities that can be detected by a NER; Mollá et al., 2007) or machine translation (Nikoulina et al., 2012). However, polysemy and synonymy, including variation of named entities (e.g. “John Smith”, “Mr Smith”, “John”), ambiguities between entities (e.g. “Ericsson” as a person vs. a company),

ambiguities between entities and common words (e.g. “Inga” as a proper noun vs. a pronoun in Swedish), nesting (e.g. “Bank of [New York]”) and several other problems, have clearly contributed to the fact that, after several decades of research in the area (Grishman and Sundheim, 1996) and the substantial efforts that have been made in the area of NER, the task is actually far from solved (Marrero et al., 2013). Nevertheless, recent NER approaches achieve high F-scores (90.8%; Ratinov and Roth, 2009) on standard data sets, such as the one developed for the CoNLL-2003 NER shared task (Tjong Kim Sang and de Meulder, 2003).

2. Named Entity Recognition for Swedish

In recent years, a small number of NER systems have also emerged in the Swedish research terrain. Kokkinakis (1998) describes a rule based system that recognizes nine entity types, both generic ones (*persons*, *organizations*, *geographical places* and *time sequences*) and also some domain dependent ones in the area of “drug enforcement” (*money expressions*, *transportation and communication means*, *social places* and *narcotic substances*). Dalianis and Åström (2001) present a NER system, SweNam, that employs rules, lexicons, and machine learning techniques for the recognition and classification of four entity types: *persons*, *locations*, *organizations* and *time*. Kokkinakis (2004) describes a further improvement and enhancement of the rule-based system described earlier, which comprises a number of modules organized into layers and applied sequentially, in a pipeline fashion.

Borin and Kokkinakis (2010) further describe the application and adaptation of the aforementioned NER system on Swedish classical literary works, mainly from the 19th and early 20th century. Salomonsson et al. (2012) give a description of yet another Swedish NER system that recognizes *persons*, *locations*, *organizations* and the

category *miscellaneous* (which incorporates entities such as *product, myth, event, animal* and *work*). Salomonsson et al.'s system relies on machine learning techniques, using standard features such as part-of-speech tags of the surrounding and the current token, the word tokens themselves, the preceding two named entity tags and some other Boolean features such as initial capitalization and if a word contains digits. Finally, Ek et al. (2011) describe yet another Swedish NER system that recognizes named entities (*locations, names, dates, times* and *telephone numbers*) in Short Message Services (SMS) written in Swedish, that runs on an Android telephones.

3. HFST-SweNER

The NER tagger described by Kokkinakis (2004) has been converted and adapted to the Helsinki Finite-State Transducer Technology (HFST) platform. This new HFST-based NER (HFST-SweNER) is a full-fledged open source implementation that uses the pattern-matching tool HFST Pmatch in the HFST toolkit (Lindén et al. 2011; 2013). HFST Pmatch (hereafter simply Pmatch) has been modeled after Karttunen's Finite-State Transducer (FST) pattern matching tool, pmatch (Karttunen, 2011). An important motivation for the work was to enable linguists who are familiar with the Xerox XFST morphology development environment to migrate their skills to writing NER and to combine NER and other language processing components in a uniform, standardized, open finite-state framework. With Pmatch, we have achieved this goal. By migrating a mature NER like the Swedish one to Pmatch we verified that the Pmatch environment is competitive with regard to expressiveness, file size, compilation and run-time speed.

4. The NER's Major Parts

The original Swedish NER system, SweNER, (Kokkinakis, 2004) is comprised of a number of modules organized into layers and applied sequentially in a pipeline fashion. The major components are lists of multiword names (approximately 4,116 bigrams and 1,504 trigram generic entities, such as *US Virgin islands* and *New York*). The lists of multiword named entities are matched directly against the text being processed since empirical evidence has shown that such n-grams are reliable and can be safely applied in the early stages of NER for nearly any document genre, since there are rarely ambiguities or conflicts between multiword entities. Entity lists can provide very high precision, but low recall; they are also at first glance intuitively quite clear, but there are a lot of grey areas caused by a plethora of linguistic phenomena that can be encountered in real texts. In many of such cases the internal and external evidence (local context) can decide an appropriate label.

Therefore, to remedy some of these problems, the next major component in the pipeline consists of sets of linguistic rules – grammars – for each type of entity recognized by the system. Each rule in such a grammar defines patterns in the entity itself and its local context (e.g., typical classes of trigger words, designators and

modifiers). These linguistic rules are used after the multiword entity look up stage is terminated.

Large lists¹ of single-word names, currently 114,508 entities (categorized according to the description provided in Section 4.1) are consulted at a later step. Each named entity in these lists is marked for a major type and a minor subtype which further specifies the entity in question. At each processing stage a theory revision and refinement filter reviews the annotations, in order to detect and resolve possible errors and assign new annotations based on existing ones. This can involve merging compatible annotations, modifying and even deleting partial annotations and annotated fragments. This module also applies a set of manually built templates, a kind of local discourse structures, for recognizing potentially new entities that are neither in a gazetteer nor recognized by the grammar component and for which the surrounding context provides a reliable base for the recognition, e.g., in enumerations. For instance, the illustrative pattern '<ENTITY₁>Qqq</ENTITY₁>, Xxx and <ENTITY₂>Zzz </ENTITY₂>' (where 'Xxx' is meant to be any token, Qqq and Zzz two previously recognized entities with labels ENTITY₁ and ₂ of the same class) will annotate the hypothetical entity Xxx with the same label as in ENTITY₁ and ₂.

4.1 Named Entity Taxonomy

As previously discussed, the nature and type of named entities vary depending on the task under investigation or the target application. In any case, *person names, location* and *organization names* are considered 'generic'. Several attempts have been made to introduce richer named entity hierarchies; (Fleischman and Hovy, 2002; Sekine, 2004). We balance these initiatives by implementing a rather fine-grained named entity taxonomy with eight main named entity types as well as several (52) subtypes (Johannesen et al., 2005; Kokkinakis, 2004). The eight main categories are:

1. **Person** (PRS): people names (forenames, surnames), animal/pet names, mythological etc.;
2. **Location** (LOC): functional, geographical, geo-political, astronomical, street names;
3. **Organization** (ORG): political, athletic, media, military, transportation, education etc.;
4. **Artifact** (OBJ): food/wine products, prizes, means of communication (vehicles), etc.;
5. **Work&Art** (WRK): printed material, names of films, novels and newspapers, sculptures, etc.;
6. **Event** (EVN): religious, athletic, scientific, cultural, races, championships, battles, etc.;
7. **Measure/Numerical** (MSR): volume, age, index, dosage, web-related, speed etc.;
8. **Temporal**² (TME).

¹ We tried to avoid building large entity lists in order to mitigate the risk of identifying a large number of false positives. Smaller gazetteers are often a wiser choice (Mikheev et al., 1999).

² Temporal expressions recognized include both relative (*nästa*

```

(a) <s id=jc04-041> Han kom till <ENAMEX TYPE="LOC" SBT="PPL">Stockholm</ENAMEX> <TIMEX TYPE="TME"
SBT="DAT">1885</TIMEX> , fick en organisatorisk bas i <ENAMEX TYPE="ORG" SBT="PLT">Socialdemokratiska
klubben</ENAMEX> ( senare <ENAMEX TYPE="ORG" SBT="PLT">Socialdemokratiska förbundet</ENAMEX> ) ,
kunde starta <ENAMEX TYPE="PRS" SBT="HUM">Social-Demokraten</ENAMEX> och fick
medarbetare som <ENAMEX TYPE="PRS" SBT="HUM">Axel Danielsson</ENAMEX> , <ENAMEX TYPE="PRS"
SBT="HUM">Fredrik Sterky</ENAMEX> och , <TIMEX TYPE="TME" SBT="DAT">året därpå</TIMEX> , <ENAMEX
TYPE="PRS" SBT="HUM">Hjalmar Branting</ENAMEX> . <s>

(b) <s id=jc04-041> Han kom till <name type=place> Stockholm </name> 1885 , fick en organisatorisk bas i <name
type=inst> Socialdemokratiska klubben </name> ( senare <name type=inst> Socialdemokratiska förbundet
</name> ) , kunde starta <name type=inst> Social-Demokraten </name> och fick medarbetare som <name
type=person> Axel Danielsson </name> , <name type=person> Fredrik Sterky </name> och , året därpå , <name
type=person> Hjalmar Branting </name> . <s>

```

Figure 1. Example sentence from the gold standard used for evaluation (see Section 7.3), annotated by the HFST-SweNER (a) and the original manual annotation (b). Note that in (a) the entity *Social-Demokraten*, is erroneously annotated as a person (PRS), which according to the manual annotation in the gold standard it should have been an *inst* (ORGanization).

Figure 1 illustrates how the named entity annotations look like in the underlying format. Identified named entities in context are surrounded by the tags `<ENAMEX3>` ... `</ENAMEX>`, entity category 1-6; `<NUMEX>` ... `</NUMEX>`, entity category 7, or `<TIMEX>` ... `</TIMEX>`, entity category 8. The starting tag also contains two attributes, which further specify the type and subtype of each recognized named entity. Thus, `<ENAMEX TYPE='ORG' SBT='PLT'>` in Figure 1 provides an example in which the recognized named entities by HFST-SweNER (a) have the main type and their subtype annotated. The manual annotated version in the gold standard (b) is given for comparison: `<s id=jc04-041> Han kom till Stockholm 1885, fick en organisatorisk bas i Socialdemokratiska klubben (senare Socialdemokratiska förbundet), kunde starta Social-Demokraten och fick medarbetare som Axel Danielsson, Fredrik Sterky och, året därpå, Hjalmar Branting <s>`, ‘`<s id=jc04-041> He came to Stockholm in 1885, had an organizational base in the Social Club (later Social Democratic Federation), could start the Social Democrat and got employees such as Axel Danielsson, Fredrik Sterky and, the following year, Hjalmar Branting. <s>`’.

The HFST-SweNER tagger, described in sections 5 and 6, identifies and classifies named entities into exactly the same set of the predefined major entity categories and their subtypes as the SweNER does.

5. Converting SweNER to HFST Pmatch

The main aim of this work was to convert the original SweNER recognition rules, written basically in the Flex lexical analyzer and other scripting languages, to Pmatch expressions as faithfully as possible. We retained the original NER pipeline structure: 24 recognition stages and a theory revision and refinement filter that modifies, removes and adds new tags based on existing ones. Each stage forms a finite-state transducer.

Since both Flex and Pmatch are based on

recognizing the leftmost longest matches of regular expressions, we were able to automate most of the conversion. A conversion script analysed the Flex actions to split a Flex regular expression pattern into a name and its context. The theory revision and refinement filter was converted by hand, since its rules were more varied than those in the recognizers. In contrast with the original SweNER, also the gazetteers needed to be compiled to finite-state automata. To facilitate that, Pmatch has a construct for reading an external file containing a list of strings.

Rule ordering in Flex was simulated by using weights in Pmatch expressions. If several Flex rules have the same leftmost longest match, the first one is chosen, so the rules can be ordered from the most specific to the most general. Although Pmatch cannot guarantee any specific order of matching for unweighted expressions, a desired order can effectively be achieved by adding to all alternative match expressions a penalty weight that is the higher the later an expression should be considered. Using weights is in general simpler than the alternative approaches that would add more detailed context conditions or subtract the more specific regular expressions from the more general. For example, the following expressions state that capitalized words ending in *gatan* should be tagged as street names (EnamexLocStr), except for *Vintergatan* ‘Milky Way’, which should remain untagged (for a brief explanation of Pmatch syntax, see Section 6):

```

define Except {Vintergatan}; 1
define Gatan UppercaseAlpha Alpha+ EndTag(EnamexLocStr); 2
define TOP Except | Gatan;

```

Without the penalty weights (the numbers following the semicolons), *Vintergatan* might be incorrectly tagged as a street name.

Differences remaining between the recognition results of the original SweNER and HFST-SweNER are explained by differences in the semantics of Flex NER rules and Pmatch. In particular, the regular expression patterns in the Flex rules cover the contexts in addition to the name to be recognized, whereas Pmatch excludes

vecka ‘next week’) and absolute expressions (*klockan 8 på morgonen i dag* ‘8 o’clock in the morning today’).

³ ENAMEX stands for ‘Extended NAMEXpression’.

contexts from its leftmost longest match. Consequently, the leftmost longest match at a certain point in text may be found by different expressions in Flex and Pmatch.

Cyclic finite-state automata which appear as part of a complex regular expression sometimes grow considerably when the expression is determinized. To reduce the size of and the compilation time required for the largest NER automata, we selectively added guards around some of the subexpressions to keep them from being determinized as part of their context. This introduces some limited non-determinism during run time but keeps the local automata and their compilation time reasonably small. The guards are called Pmatch insertion statements. Using insertion statements reduced the size and compilation time by a factor of 5 on the average and by a factor of 40 in the best case. The largest improvements were in cases where the insertion statement was added to recurring sub-expressions recognizing general patterns, such as “any word”, at the beginning of a recognition rule.

6. NE Recognition with HFST Pmatch

A key feature of Pmatch, making it well-suited for NER, is the ability to efficiently add XML-style tags around substrings matching a regular expression, as in Karttunen (2011). For example, the expressions in Figure 2 mark capitalized words with `EnamexOrgCrp` only if they are preceded by the string *rörelseresultatet för* ‘operating profit of’:

```
define NoSpTag [? - [Whitespace]"<">"];
define CapWord2 UppercaseAlpha NoSpTag+;
define OrgCrpOpProfit CapWord2 [ " " CapWord2]*
EndTag(EnamexOrgCrp) LC({Rörelseresultatet för });
define TOP OrgCrpOpProfit;
```

Figure 2: Pmatch example.

E.g.: *rörelseresultatet för* `<EnamexOrgCrp>Comp Systems</EnamexOrgCrp>`⁴ ...’. Pmatch considers leftmost longest matches of the expression named TOP in the input and adds the tags specified with `EndTag` (`TagName`) in TOP or in the expressions to which TOP refers. To disambiguate between matches, a regular expression may be accompanied with a context condition specifying that a match should be considered only if the left or right context of the match matches the context condition (specified in `LC()` or `RC()`, respectively). The built-in set *Whitespace* denotes any whitespace character and *UppercaseAlpha* any uppercase letter.

7. Evaluation and Gold Standard

For the evaluation we used an existing Swedish gold

⁴ Pmatch currently only allows simple tags without attributes. These tags are converted to those of the original SweNER in a post-processing stage, so that `<EnamexOrgCrp>Comp Systems</EnamexOrgCrp>` becomes `<ENAMEX TYPE="ORG" SBT="CRP">Comp Systems</ENAMEX>`.

standard corpus, namely the Stockholm-Umeå Corpus, SUC version 3.0 (SUC3.0; Gustafson-Capková and Hartmann, 2006). In SUC3.0, all named entities (34,194) have been manually annotated and could thus be used for the gold standard evaluation. Table 1 shows the entities found in SUC 3.0 and their corresponding labels before and after their conversion.

Although SUC’s entity labels are not 100% compatible with HFST-SweNER’s tagset, it is the closest we can get with respect to a gold standard corpus for automatic evaluation of the HFST-SweNER. The automatic evaluation was made with the *conllev* script.⁵

SUC3.0	#	HFST-SweNER
person	15,128	PRS
place	8,776	LOC
inst	6,334	ORG
work	1,887	WRK
product	638	OBJ
other	542	???
animal	364	PRS/ANM
myth	280	PRS/MTH
event	245	EVN
num	18,098	???

Table 1. Named entity tags and their occurrences in SUC3.0, and their conversion to the HFST-SweNER.

We evaluated the NER in two ways. Firstly, we conducted a functional evaluation (Section 7.2), secondly an evaluation based on all entities (with minor exceptions, see Section 7.1) found in the same gold standard (Section 7.3).

7.1 Preprocessing

In order to ease the comparison between the annotations, we harmonized the annotations between the original SUC3.0 and the ones returned by HFST-SweNER. Thus in order to make the evaluation more reliable, we performed a few preprocessing steps in SUC3.0. For convenience reasons, we converted the SUC3.0 label tags (i.e., *name type=*) to the ones used in HFST-SweNER (i.e., *ENAMEX*). The most important preprocessing step was applied for the category “person” for which SUC3.0 sometimes includes in the annotation an apposition or designator, often a common noun in lower case. These, however, seem to be used in an ad hoc manner; for instance in SUC3.0 there are annotations such as: `<name type=person>kusin Bosse</name>` ‘cousin Bosse’ (SUC3.0-file: kr06); `Att <name type=person>morbror Ture </name> erbjudit er tio procent [...]` ‘That uncle Ture offered you ten percent ...’ (SUC3.0-file: kr06) or `<name type=person>Tant Sigrid</name>` ‘Aunt Sigrid’ (SUC3.0-file: kr04). Therefore, these designators were moved outside of the annotation. Thus, the three previous examples were changed to: `kusin <name`

⁵ For the evaluation we used the *conllev* script (v. 04-01-26) by Tjong Kim Sang `<http://www.cnts.ua.ac.be/conll2002/ner/bin/conllev.txt>`.

type=person>Bosse</name> and *Att morbror <name type=person> Ture </name> erbjudit er tio procent [...]*.

Furthermore, annotations enclosed in “<num>” in SUC3.0 were not considered at all since these were not classified in a homogeneous manner. These annotations involved a large variety of numerical information of various category types, such as temperature or volume. Note that HFST-SweNER can recognize various different numerical types of information that, in conjunction with local context, decides an appropriate category. Moreover, the SUC3.0 category “<other>” was also not used for similar reasons. In the case of the HFST-SweNER most of the entities in SUC3.0 annotated as “<other>” could also be captured, but as part of the various categories; for instance: “object”, such as names of transportation means, vessels (SUC3.0-file kl05): *<name type=other> M/S Buchanan </name>*; “event” (SUC3.0-file kr02) such as: *Kampanjen <name type=other> ANNONSERA FÖR FRED </name> [...]* ‘The campaign ADVERTISE FOR PEACE’ or “functional location” (SUC3.0-file kr01a): *[...] fylla <name type=other> Konserthusets </name> stora sal [...]* fill the Concert hall’s great hall’.

7.2 Functional Evaluation

We evaluated the correctness of HFST-SweNER by comparing its recognition result with that of SweNER on (a large part of) the Swedish EuroParl v6 corpus.⁶ The corpus consisted of over 37 million words with about a million recognized named entities. The recall of HFST-SweNER with regard to SweNER was 99.1% and the precision 98.3%. HFST-SweNER recognized 0.9% additional names, and missed 0.1% of the names recognized by SweNER; 0.7% were recognized at the same position but as longer than by SweNER and 0.1% as shorter; and 0.03% were assigned a different type. The differences in the results were mostly due to the differences between the matching semantics of Flex and Pmatch, and they can be corrected by modifying the HFST-SweNER Pmatch rules.

7.3 Evaluation Based on the Gold Standard

Using SUC3.0 the generic entity types can be (almost) safely evaluated, e.g. *person*, *location* (annotated as *place* in SUC3.0) and *organization* (annotated as *inst* in SUC3.0). There is a general consensus of the content of these classes, and since SUC3.0 doesn’t make a further finer-grained sub-classification of the entities, e.g. between political and sports related organizations as in SweNER, the evaluation can be effectively conducted. For the rest of the entity classes in SUC3.0, *event* and *work* remained unchanged, while *product* was mapped to *object* and *animal* and *myth* to appropriate subtypes in the person class.

Table 2 shows the results of this evaluation, which illustrates some of the difficulties in the recognition, particularly, between the location and organization classes. For instance, the low recall for the ORG class can be

explained by the fact that SUC3.0 does not mark all occurrences of certain entities such as: *Naturskyddsföreningen* ‘the Swedish Society for Nature Conservation’, *Socialstyrelsen* ‘The National Board of Health and Welfare’ or *Naturvårdsverket* ‘the Swedish Environmental Protection Agency’ or not at all other entities such as *Socialnämnden* ‘Social Services Committee’. The results in Table 2 are shown with and without considering whether an identified entity is part of a multi token annotation. Using the so called IOB representation, each identified named entity is prefixed by either ‘B-’ or ‘I-’. Here ‘B’ (for Begin) marks the first word in a multi token named entity (or a single entity token), while further named entity tokens receive the prefix ‘I’, for Inside. In text, tokens that are not part of named entity receive the annotation O (for Outside).

Label	Precision	Recall	FB1	#
PRS	91.34%	91.32%	91.33%	22492
LOC	73.03%	83.78%	78.04%	10917
ORG	70%	43.17%	53.41%	5220
<i>B-PRS</i>	88.78%	87.45%	88.11%	15519
<i>I-PRS</i>	87.90%	90.89%	89.37%	6973
<i>B-LOC</i>	75.42%	84.69%	79.79%	9854
<i>I-LOC</i>	43.37%	62.21%	51.11%	1063
<i>B-ORG</i>	65.00%	39.37%	49.04%	3837
<i>I-ORG</i>	76.28%	49.53%	60.06%	1383
OBJ	64.14%	33.12%	43.68%	488
WRK	52.91%	18.56%	27.48%	1544
EVN	41.62%	66.85%	51.30%	591
<i>B-OBJ</i>	55.23%	29.78%	38.70%	344
<i>I-OBJ</i>	78.47%	36.81%	50.11%	144
<i>B-EVN</i>	42.97%	66.12%	52.09%	377
<i>I-EVN</i>	35.98%	62.60%	45.70%	214
<i>B-WRK</i>	44.88%	17.44%	25.11%	733
<i>I-WRK</i>	46.49%	14.99%	22.67%	811
All	79.02%	70.56%	74.55%	

Table 2. Evaluation results for the entity types in SUC3.0; ‘#’ occurrences in SUC3.0 after the preprocessing steps (see Section 7.1)

8. Conclusions

In this paper we have described the conversion, modeling, adaptation and evaluation of a Swedish named entity recognizer to the Helsinki Finite-State Transducer Technology toolkit intended for various language software applications. Language technology in the form of named entity recognition and annotation systems can be usefully applied for building computer lingware to support various applications e.g., in the area of digital humanities. For instance, entity annotations allow a more semantically-oriented exploration of textual content in large digitized archives (Borin and Kokkinakis, 2010; Elson et al., 2010).

What can we conclude from the results of the evaluation? Apparently the results do not reflect the potential of the HFST-SweNER in a fair way. Several entity annotations in SUC3.0 are not consistent and there are various encountered examples that supported the fact

⁶ <<http://www.statmt.org/europarl/archives.html>>.

that the evaluation results should be taken with caution since there is a strong indication that these inconsistencies have a negative impact on the evaluation performance and the figures in Table 2. Consider for instance the examples: `<s id=ce02b-002> <name type=inst> Konserthusets </name> stora sal: [...] <s>` and `<s id=kr01a-035> [...] fylla <name type=other> Konserthusets</name> stora sal. <s>`, the first annotated as *inst* (organization) and the second as *other*; both, however, refer to a physical location ‘Concert hall’. Certain pieces of work as well as events are not marked in SUC3.0, such as Catechesationsbok ‘Catechization book (SUC3.0-file jd06); *Valborgsmässoafton* ‘Walpurgis Night’; or *Kristi himmelfärd* ‘Ascension’; while other similar kind of entities are sometime marked sometime not; e.g. (SUC3.0-file kk11) `<name type=other> Svenska flaggans dag </name>` ‘the Swedish flag day’ but not the *Skånska flaggans dag* ‘Scanian flag day’ (in SUC3.0-file af05). Other multiword entities are also annotated in an ad hoc manner, for instance, only the first part of the expression, i.e. ‘Big’, is annotated in the following example `<name type=event> Big </name> Bang-teorins` ‘the Big Bang theory’s’ (SUC3.0-file: fh12). Therefore, we believe that the gold standard we used for the evaluation needs to be thoroughly re-checked for the entity annotations it contains. Of course, this fact does not diminish the importance of SUC3.0 as a very valuable resource for evaluating Swedish NER components.

Although a whole new NER system could be rebuilt in a machine learning framework or other rule-based framework, we decided to use Pmatch in order to verify that we are offering an environment that is competitive with the technologies used in SweNER with regard to expressiveness, compilation speed, file size and run-time speed. Note also that although we are relatively satisfied with the first three, we still need to work on the run-time speed in comparison with Flex. We believe that for even better results we need to integrate the strengths of rule-based systems with the ones provided by supervised learning, a task we have left for exploration in the near future.

9. Acknowledgements

This work was supported by the Centre for Language Technology <http://clt.gu.se/> at the University of Gothenburg and the FIN-CLARIN at the University of Helsinki www.helsinki.fi/finclarin. The current version of the HFST-SweNER is available from <http://clt.gu.se/clt-toolkit>. The license for HFST is LGPL by default but some of HFST’s runtime libraries are available with an Apache license as well. The HFST-SweNER includes the uncompiled and compiled rule set with a CC-BY license which is also the license for the HFST-SweNER.

10. References

Ananiadou S., Friedman C. and Tsujii J. (2004). Introduction: named entity recognition in biomedicine. *Journal of Biomedical Informatics*. 37(6): 393-395.

- Borin L. and Kokkinakis D. (2010). Literary Onomastics and Language Technology. In *Literary Education and Digital Learning. Methods and Technologies for Humanities Studies*. van Peer W., Zyngier S. and Viana V. (eds). Pp. 53-78. IGI Global.
- Dalianis H. and Åström E. (2001). SweNam – a Swedish named entity recognizer. Technical Report. Department of Numerical Analysis and Computing Science, TRITA-NA-P0113 - IPLab-189. Stockholm, Sweden. <http://ftp.nada.kth.se/IPLab/TechReports/IPLab-189.pdf>
- Dozier C., Kondadadi R., Light M., Vachher A., Veeramachaneni S. and Wudali R. (2009). Named Entity Recognition and Resolution in Legal Text. In *Semantic Processing of Legal Texts*. E. Francesconi, S. Montemagni, W. Peters, D. Tiscornia, eds. Springer Verlag.
- Ek T., Kirkegaard C., Jonsson H. and Nugues P. (2011). Named Entity Recognition for Short Text Messages. *Procedia - Social and Behavioral Sciences*. Special Issue on Computational Linguistics and Related Fields (Proceedings of the Pacific Association for Computational Linguistics, PACLING). Volume 27, pages 178-187. Elsevier.
- Elson K. D., Dames N. and McKeown R. K. (2010). Extracting Social Networks from Literary Fiction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Pages 138-147. Uppsala, Sweden.
- Fleischman M. and Hovy E. (2002). Fine Grained Classification of Named Entities. In *Proceedings of the 19th International Conference on Computational Linguistics*. Pages 1–7. Taipei.
- Giuliano C. and Gliozzo A. (2008). Instance-Based Ontology Population Exploiting Named-Entity Substitution. Proceedings of the 22nd International Conference on Computational Linguistics (COLING). Pages 265–272. Manchester.
- Grishman R. and Sundheim B. (1996). Message Understanding Conference - 6: A Brief History. *Proceedings of the 16th conference on Computational Linguistics (COLING)*. Vol. 1, pages 466-471. Denmark.
- Gustafson-Capková S. and Hartmann B. (2006). *Manual of the Stockholm Umeå Corpus version 2.0*. Description of the content of the SUC 2.0 distribution, including the unfinished documentation by Gunnel Källgren. <http://spraakbanken.gu.se/parole/Docs/SUC2.0-manual.pdf>
- Johannessen J. B., Hagen K., Haaland Å., Björk Jónsdóttir A., Nøklestad A., Kokkinakis D., et al. (2005). Named entity recognition for the mainland Scandinavian languages. *Literary and Linguistic Computing*. 20 (1): 91–102.
- Karttunen L. (2011). Beyond morphology: Pattern matching with FST. In *Systems and Frameworks for Computational Morphology. Second International Workshop, (SFCM)*. Proceedings, vol. 100 of Communications in Computer and Information Science. Mahlow C. and Piotrowski M. (eds). pages 1–13, Berlin Heidelberg, Springer.

- Kokkinakis D. (1998). AVENTINUS, GATE and Swedish Lingware. *Proceedings of the 11th NODALIDA Conference*. Copenhagen, Denmark.
- Kokkinakis D. (2004). Reducing the Effect of Name Explosion. *Proceedings of the LREC Workshop: Beyond Named Entity Recognition, Semantic labelling for NLP tasks. Fourth Language Resources and Evaluation Conference (LREC)*. Pp. 1-6. Lissabon, Portugal.
- Kumaran G. and Allan J. (2004). Text Classification and Named Entities for New Event Detection. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR)*. Pages 297–304. Sheffield, UK.
- Lindén K., Axelson E., Drobac S., Hardwick S., Kuokkala J., Niemi J., Pirinen T. A. and Silfverberg M. (2013). HFST — a system for creating NLP tools. In *Systems and Frameworks for Computational Morphology, Third International Workshop, SFCM*. Proceedings, vol. 380 of Communications in Computer and Information Science. Mahlow C. and Piotrowski M. (eds). Pages 53–71. Berlin Heidelberg. Springer.
- Lindén K., Axelson E., Hardwick S., Pirinen T. A. and Silfverberg M. (2011). HFST — framework for compiling and applying morphologies. In *Systems and Frameworks for Computational Morphology, Second International Workshop, (SFCM)*. Proceedings, vol. 100 of Communications in Computer and Information Science. Mahlow C. and Piotrowski M. (eds). Pages 67–85. Berlin Heidelberg. Springer.
- Marrero M., Urbano J., Sánchez-Cuadrado S., Morato J. and Gómez-Berbís J.M. (2013). Named Entity Recognition: Fallacies, Challenges and Opportunities. *Journal of Computer Standards & Interfaces*. Vol. 35:5, pages 482–489. Elsevier.
- Mikheev A., Moens M. and Grover C. (1999). Named entity recognition without gazetteers. In *Proceedings of the European ACL*. Bergen, Norway.
- Mollá D., van Zaanen M. and Cassidy S. (2007). Named Entity Recognition in Question Answering of Speech Data. *Proceedings of the Australasian Language Technology Workshop*. Pages 57-65. Melbourne, Australia
- Nikoulina V., Sandor A. and Dymetman M. (2012). Hybrid Adaptation of Named Entity Recognition for Statistical Machine Translation. *Second MLAHMT Workshop (a COLING workshop)*. Pages 1–16. Mumbai.
- Ratinov L. and Roth D. (2009). Design Challenges and Misconceptions in Named Entity Recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL)*. Pages 147–155, Boulder, Colorado.
- Salomonsson A., Marinov S. and Nugues P. (2012). Identification of Entities in Swedish. In *Proceedings of the Swedish Language Technology Conference*. Lunds, Sweden.
- Sekine S. (2004). Definition, dictionaries and tagger for extended named entity hierarchy. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*. Lisbon, Portugal.
- Sekine S. and Ranchhod E. (eds). (2009). *Named Entities: Recognition, classification and use*. John Benjamins Publishing
- Suchanek F. M., Ifrim G. and Weikum G. (2006). LEILA: Learning to Extract Information by Linguistic Analysis. *Proceedings of the 2nd Workshop on Ontology Learning and Population*. Pages 18–25, Sydney. <<http://acl.ldc.upenn.edu/W/W06/W06-0503.pdf>>
- Tjong Kim Sang E.F. and de Meulder F. (2009). Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In: *Proceedings of CoNLL-2003*. Paes 142-147. Edmonton, Canada
- Wang Y. (2009). Annotating and Recognising Named Entities in Clinical Notes. In *Proceedings of the ACL-IJCNLP Student Research Workshop*. Pages 18–26. Suntec, Singapore.