DKIE: Open Source Information Extraction for Danish

Leon Derczynski
University of Sheffield
leon@dcs.shef.ac.uk

Camilla Vilhelmsen Field
University of Southern Denmark
cafie13@student.sdu.dk

Kenneth S. Bøgh
Aarhus University
ksb@cs.au.dk

Abstract

Danish is a major Scandinavian language spoken daily by around six million people. However, it lacks a unified, open set of NLP tools. This demonstration will introduce DKIE, an extensible open-source toolkit for processing Danish text. We implement an information extraction architecture for Danish within GATE, including integrated third-party tools. This implementation includes the creation of a substantial set of corpus annotations for data-intensive named entity recognition. The final application and dataset is made are openly available, and the part-of-speech tagger and NER model also operate independently or with the Stanford NLP toolkit.

1 Introduction

Danish is primarily spoken in the northern hemisphere: in Denmark, on the Faroe islands, and on Greenland. Having roots in Old Norse, Danish bears similarities to other Scandinavian languages, and shares features with English and German.

Previous tools and language resources for Danish have suffered from license restrictions, or from using small or non-reusable datasets. As a result, it is often difficult to use Danish language technologies, if anything is available at all. In cases where quality tools are available, they often have disparate APIs and input/output formats, making integration time-consuming and prone to error.

To remedy this, this paper presents an open-source information extraction toolkit for Danish, using the established and flexible GATE text processing platform (Cunningham et al., 2013). To this end, there are three main goals:

Adaptation: The application adapts to colloquial and formal Danish.

Interoperability: DKIE is internally consistent and adopts unified, well-grounded solutions to the problems of processing Danish. Where possible, DKIE re-uses existing components, and strives for compatibility with major text processing architectures.

Portability: It is preferable for developed components to be readily movable within the chosen architecture, GATE, and without, usable independently.

Openness: The resultant application, and corpora and annotations developed in its creation, are as freely-available as possible.

The remainder of this paper first discusses considerations specific to the language and prior work, then introduces the information extraction pipeline, followed by an evaluation of the tools provided.

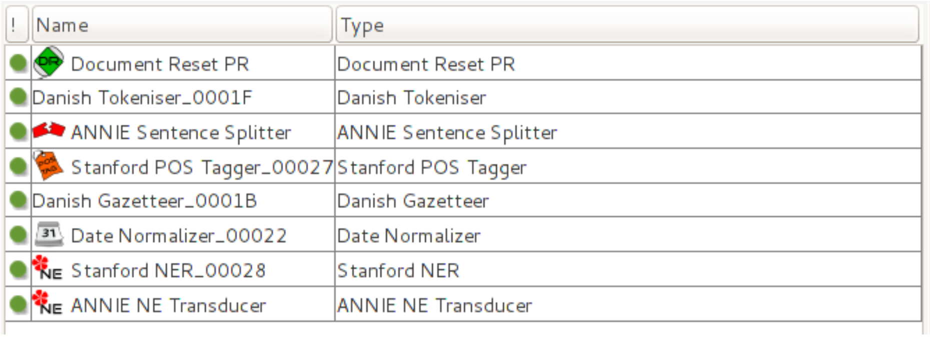
2 Processing Danish

There are a few representational issues for Danish that are not solved in a unified fashion across existing technological issues. DKIE builds upon major standards in general linguistic annotation and in Danish to unify these solutions.

Danish is written using the Latin alphabet, with the addition of three vowels: æ, ø and å, which may be transliterated as ae, oe and aa respectively. It is similar to English in terms of capitalisation rules and character set.

Over time, the orthography of Danish has shifted. Among other things, a spelling reform in 1948 removed the capitalisation of nouns, and introduced the three vowel characters to represent existing vowel digraphs. There were also spelling shifts in this reform (e.g. *kjærlighed* to *kærlighed*). In addition, some towns and municipalities have changed the spelling of their name. For example, Denmarks second-largest city Aarhus changed its name to Århus with the 1948

Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics, pages 61–64,
Gothenburg, Sweden, April 26-30 2014. ©2014 Association for Computational Linguistics



!	Name	Type
●	Document Reset PR	Document Reset PR
●	Danish Tokeniser_0001F	Danish Tokeniser
●	ANNIE Sentence Splitter	ANNIE Sentence Splitter
●	Stanford POS Tagger_00027	Stanford POS Tagger
●	Danish Gazetteer_0001B	Danish Gazetteer
●	Date Normalizer_00022	Date Normalizer
●	Stanford NER_00028	Stanford NER
●	ANNIE NE Transducer	ANNIE NE Transducer

Figure 1: The ANNIE-based information extraction pipeline for Danish

reform, although Aalborg and Aabenraa did not. Later, in 2011, the city reverted from Århus to Aarhus. The city's university retained the Aarhus spelling throughout this period.

The effect of these relatively recent changes is that there exist digitised texts using a variety of orthographies not only to represent the same sound, as also in English, but also the same actual word. A language processing toolkit for Danish must exhibit sensitivity to these variances.

In addition, Danish has some word boundary considerations. Compound nouns are common (e.g. *kvindehåndboldlandsholdet* for "the women's national handball team"), as are hyphenated constructions (*fugle-fotografering* for "bird photography") which are often treated as single tokens.

Finally, abbreviations are common in Danish, and its acronyms can be difficult to disambiguate without the right context and language resource (e.g. OB for *Odense Boldklub*, a football club).

3 Background

The state of the art in Danish information extraction is not very interoperable or open compared to that for e.g. English. Previous work, while high-performance, is not available freely (Bick, 2004), or domain-restricted.[1] This makes results difficult to reproduce (Fokkens et al., 2013), and leads to sub-optimal interoperability (Lee et al., 2010). Even recent books focusing on the topic are heavily licensed and difficult for the average academic to access. Further, prior tools are often in the form of discrete components, hard to extend or to integrate with other systems.

Some good corpus resources are available, most recently the Copenhagen Dependency Treebank (CDT) (Buch-Kromann and Korzen, 2010), which built on and included previously-released corpora for Danish. This 200K-token corpus is taken from news articles and editorials, and includes document structure, tokenisation, lemma, part-of-speech and dependency relation information.

The application demonstrated, DKIE, draws only on open corpus resources for annotation, and the annotations over these corpora are released openly. Further, the application is also made open-source, with each component having similar or better performance when compared with the state-of-the-art.

4 Information Extraction Pipeline

This section details each step in the DKIE pipeline. A screenshot of the tool is shown in Figure 1.

4.1 Tokeniser

We adopt the PAROLE tokenisation scheme (Keson and Norling-Christensen, 1998). This makes different decisions from Penn Treebank in some cases, concatenating particular expressions as single tokens. For example, the two word phrase *i alt* – meaning *in total* – is converted to the single token `i_alt`. A set list of these group formations is given in the Danish PAROLE guidelines.

Another key difference is in the treatment of quoted phrases and hyphenation. Phrases connected in this way are often treated as single tokens. For example, the phrase *"Se og hør"-læserne* (*readers of "See and Hear"*, a magazine) is treated as a single token under this scheme.

4.2 Part-of-Speech tagger

We use a machine-learning based tagger (Toutanova et al., 2003) for Danish part-of-speech labelling. The original PAROLE

[1] E.g. CST's non-commercial-only anonymisation tool, at http://cst.dk/online/navnegenkender/

Tagger	Token accuracy %	Sentence acc. %
DKIE	95.3	49.1
TnT	96.2	39.1

Table 1: Part-of-speech labelling accuracy in DKIE

scheme introduces a set of around 120 tags, many of which are used only rarely. The scheme comprises tags built up of up to nine features. These features are used to describe information such as case, degree, gender, number, possessivity, reflexivity, mood, tense and so on (Keson and Norling-Christensen, 1998).

The PAROLE data includes morphological encoding in tags. We separate this data out in our corpus, adding morphological features distinct from part-of-speech data. This data may then be used by later work to train a morphological analyser, or by other tools that rely on morphological information.

We combine PAROLE annotations with the reduced tagset employed by the Danish Dependency Treebank (DDT) (Kromann, 2003). This has 25 tags. We adapted the tagger to Danish by including internal automatic mapping of æ, ø and å to two-letter diphthongs when both training and labelling, by adding extra sets of features for handling words and adjusting our unknown word threshold to compensate for the small corpus (as in Derczynski et al. (2013)), and by specifying the closed-class tags for this set and language. We also prefer a CRF-based classifier in order to get better whole-sequence accuracy, providing greater opportunities for later-stage tools such as dependency parsers to accurately process more of the corpus.

Results are given in Table 1, comparing token- and sentence-level accuracy to other work using the DDT and the TnT tagger (Brants, 2000). State-of-the-art performance is achieved, with whole-sentence tagging accuracy comparable to that of leading English taggers.

4.3 Gazetteers

High precision entity recognition can be achieved with gazetteer-based named entity recognition. This is a low-cost way of quickly getting decent performance out of existing toolkits. We include two special kinds of gazetteer for Danish. Firstly, it is important to annotation the names of entities specific to Denmark (e.g. Danish towns). Secondly, entities outside of Denmark sometimes have different names specific to the Danish language (e.g. *Lissabon* for *Lisboa* / *Lisbon*).

```
id      expression          interpretation
--      ----------          --------------
 3      igaa                ADD(DCT,day,-1)
13      Num._jul            ADD(DATE_MONTH_DAY(DCT, 12, 24),
                             day, TOKEN(0))
```

Figure 2: Example normalisation rules in TIMEN. "DCT" refers to the document creation time.

As well as a standard strict-matching gazetteer, we include a "fuzzy" gazetteer specific to Danish that tolerates vowel orthography variation and the other changes introduced in the 1948 spelling reform. For locations, we extracted data for names of Danish towns from DBpedia and a local gazetteer, and from Wikipedia the Danish-language versions of the world's 1 000 most populous cities. For organisations, we used Wikipedia cross-language links to map the international organisations deemed notable in Wikipedia to their Danish translation and acroynm (e.g. the United Nations is referred to as *FN*). The major Danish political parties were also added to this gazetteer. For person names, we build lists of both notable people,[2] and also populated GATE's first and last name lists with common choices in Denmark.

4.4 Temporal Expression Annotation

We include temporal annotation for Danish in this pipeline, making DKIE the first temporal annotation tool for Danish. We follow the TimeML temporal annotation standard (Pustejovsky et al., 2004), completing just the TIMEX3 part.

Danish is interesting in that it permits flexible temporal anchors outside of reference time (Reichenbach, 1947) and the default structure of a calendar. For example, while in English one may use numbers to express a distance in days (*two days from now*) or into a month (*the second of March*), Danish permits these offsets from any agreed time. As a result, it is common to see expressions of the form *2. juledag*, which in this case is *the second christmas day* and refers to 26^{th} December.

For this pipeline, we use finite state transducers to define how Danish timexes may be recognised. We then use the general-purpose TIMEN (Llorens et al., 2012) timex normalisation tool to provide calendar or TIMEX3 values for these expressions. Example rules are shown in Figure 2.

[2] See https://en.wikipedia.org/wiki/List_of_Danes, minus musicians due to stage names

4.5 Named entities

In addition to gazetteers, we present a machine learning-based approach to entity recognition and classification in Danish. We annotated the Copenhagen Dependency Treebank for person, location and organisation entities, according to the ACE guidelines (or as close as possible). This led to a total of 100 000 extra tokens annotated for NEs in Danish, doubling the previously-available amount. We used three annotators, achieving inter-annotator agreement of 0.89 on the first 100 000 tokens; annotation is an ongoing effort.

The data was used to learn a model tuned to Danish with an existing NER tool (Finkel et al., 2005). We removed word shape conjunctions features from the default configuration in an effort to reduced sensitivities introduced by the group noun tokenisation issue. This model, and the Stanford NER tool, were then wrapped as a GATE processing resource, contributing general-purpose Danish NER to the toolkit.

5 Conclusion

We will demonstrate a modern, interoperable, open-source NLP toolkit for information extraction in Danish. The released resources are: a GATE pipeline for Danish; tools for temporal expression recognition and normalisation for Danish; part-of-speech and named entity recognition models for Danish, that also work in the Stanford NLP architecture; and named entity corpus annotations over the Copenhagen Dependency Treebank.

Acknowledgments

This work was supported by EU funding under grant FP7-ICT-2013-10-611233, Pheme, and grant agreement No. 296322, AnnoMarket. We are grateful to Anders Søgaard of Copenhagen University for comments on an earlier draft and kind help with gazetteers. The first author would also like to thank Aarhus University for their kind provision of research facilities.

References

E. Bick. 2004. A named entity recognizer for Danish. In *Proceedings of LREC*.

T. Brants. 2000. TnT: a statistical part-of-speech tagger. In *Proceedings of the sixth conference on Applied natural language processing*, pages 224–231. ACL.

M. Buch-Kromann and I. Korzen. 2010. The unified annotation of syntax and discourse in the Copenhagen Dependency Treebanks. In *Proceedings of the Fourth Linguistic Annotation Workshop*, pages 127–131. ACL.

H. Cunningham, V. Tablan, A. Roberts, and K. Bontcheva. 2013. Getting More Out of Biomedical Documents with GATE's Full Lifecycle Open Source Text Analytics. *PLoS computational biology*, 9(2):e1002854.

L. Derczynski, A. Ritter, S. Clark, and K. Bontcheva. 2013. Twitter Part-of-Speech Tagging for All: Overcoming Sparse and Noisy Data. In *Proceedings of Recent Advances in Natural Language Processing*. Association for Computational Linguistics.

J. R. Finkel, T. Grenager, and C. Manning. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370. ACL.

A. Fokkens, M. van Erp, M. Postma, T. Pedersen, P. Vossen, and N. Freire. 2013. Offspring from reproduction problems: What replication failure teaches us. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 1691–1701. Association for Computational Linguistics.

B. Keson and O. Norling-Christensen. 1998. PAROLE-DK. *The Danish Society for Language and Literature*.

M. T. Kromann. 2003. The Danish Dependency Treebank and the DTAG treebank tool. In *Proceedings of the Second Workshop on Treebanks and Linguistic Theories*, page 217.

K. Lee, L. Romary, et al. 2010. Towards interoperability of ISO standards for Language Resource Management. *Proc. ICGL 2010*.

H. Llorens, L. Derczynski, R. J. Gaizauskas, and E. Saquete. 2012. TIMEN: An Open Temporal Expression Normalisation Resource. In *LREC*, pages 3044–3051.

J. Pustejovsky, B. Ingria, R. Sauri, J. Castano, J. Littman, and R. Gaizauskas. 2004. The Specification Language TimeML. In *The Language of Time: A Reader*, pages 545–557. Oxford University Press.

H. Reichenbach. 1947. The tenses of verbs. In *Elements of Symbolic Logic*. Macmillan.

K. Toutanova, D. Klein, C. D. Manning, and Y. Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 173–180. ACL.