CHAPTER 7

TTNWW to the Rescue: No Need to Know How to Handle Tools and Resources

Marc Kemps-Snijders[a], Ineke Schuurman[b], Walter Daelemans[c], Kris Demuynck[d], Brecht Desplanques[d], Véronique Hoste[d], Marijn Huijbregts[e], Jean-Pierre Martens[d], Hans Paulussen[b], Joris Pelemans[b], Martin Reynaert[e,g], Vincent Vandeghinste[b], Antal van den Bosch[e], Henk van den Heuvel[e], Maarten van Gompel[e], Gertjan van Noord[f] and Patrick Wambacq[b]

[a]Meertens Instituut Amsterdam, [b]KU Leuven, [c]Universiteit Antwerpen, [d]Universiteit Gent, [e]Radboud Universiteit Nijmegen, [f]Universiteit Groningen, [g]Tilburg University

ABSTRACT

'But I don't know how to work with [name of tool or resource]' is something one often hears when researchers in Human and Social Sciences (HSS) are confronted with language technology, be it written or spoken, tools or resources. The TTNWW project shows that these researchers do not need to be experts in language or speech technology, or to know all kinds of details about the tools involved. In principle they only need to make clear what they want to achieve.

In this chapter we describe a series of tools that are already available as a webservice. Details are not presented — interested readers are referred to the papers mentioned in the References and to the TTNWW website.

7.1 Introduction

The idea behind the Flemish/Dutch CLARIN project TTNWW[1] ('**T**ST **T**ools voor het **N**ederlands als **W**ebservices in een **W**orkflow', or 'NLP Tools for Dutch as Web services in a Workflow') was that many end users of resources and tools offered by CLARIN will not know how to use them, just as they will not know where they are located. With respect to the location, the CLARIN policy is that the Human and Social Sciences (HSS) researcher does not need to know this as the infrastructure will take care of that: the only thing the user needs to do is to indicate what (s)he is interested in.

[1] https://dev.clarin.nl/node/1964.

How to cite this book chapter:

Kemps-Snijders, M, Schuurman, I, Daelemans, W, Demuynck, K, Desplanques, B, Hoste, V, Huijbregts, M, Martens, J-P, Paulussen, H, Pelemans, J, Reynaert, M, Vandeghinste, V, van den Bosch, A, van den Heuvel, H, van Gompel, M, van Noord, G and Wambacq, P. 2017. TTNWW to the Rescue: No Need to Know How to Handle Tools and Resources. In: Odijk, J and van Hessen, A. (eds.) *CLARIN in the Low Countries*, Pp. 83–93. London: Ubiquity Press. DOI: https://doi.org/10.5334/bbi.7.

The same should hold for the use of tools and resources: users do not need to know which (other) tools are to be used in order to obtain the data one is looking for. Once more, the infrastructure has to take care of that.

For the Dutch language TTNWW served as a pilot project, trying to provide this service for a whole range of existing resources (both text and speech) and tools. The envisaged end users in TTNWW were researchers in social history, literary onomastics and archaeology. Of course, the web service can also be useful for researchers in other domains, such as linguistics, media, political sciences, communication technology, and sociology. Currently, the requirements are that the resources be in Dutch (spoken or written).[2]

Once resources have been handled by (some of) the services described below, it becomes much easier for researchers to find the data they are looking for, especially for audio resources where the gain of time can be tremendous, that is if something can be found at all without the data being transcribed. Suppose one needs data about 'lead paint', nowadays considered hazardous but commonly used in the past. In metadata such a concept will only be mentioned when the document is about lead paint, not when artists are discussed and remarks about the paint they commonly used are made in passing. A specific document about, say, Rembrandt could easily escape notice, while it contains just the data one is looking for. When the transcription and the original resource are time-synchronous, the user can listen to the parts of the resource (s)he is interested in. In originally written documents it is easier to find such data once a resource is available in machine-readable format, but even in such cases the gain of time can be huge as one can search in a much more goal-oriented manner.

As shown in Figure 7.1, two main types of input are possible in TTNWW: written or spoken. The transcribed audio resources can be used as such, or they can be inserted in the pipeline for written texts.

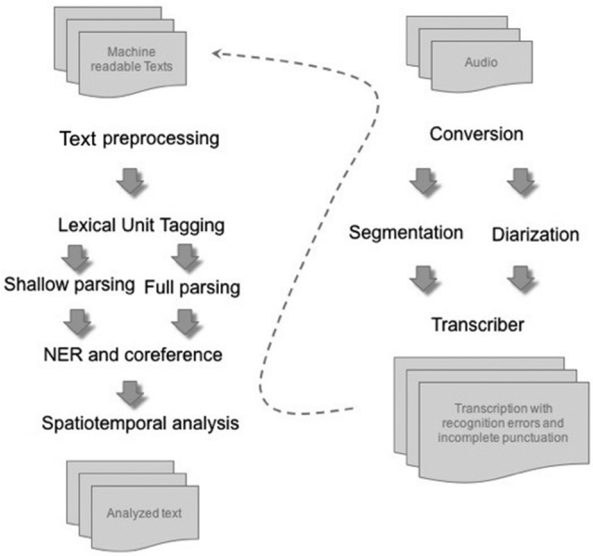

Figure 7.1: Architecture of TTNWW.

[2] But see the section on Alignment.

In the following sections we will first discuss the workflow for written texts, followed by the workflow for audio recordings. In the remainder of the chapter the TTNWW web service will be explained.

7.1.1 Formats and Web Service Support

Some necessary conditions for building a text workflow based on existing linguistic tools are that the tools need to be able to communicate and that they need to share a particular text annotation format rich enough to accommodate all the components in the workflow. FoLiA (Format for Linguistic Annotation), cf. van Gompel and Reynaert (2013), was explicitly developed to this end in the scope of both TTNWW and other projects. The format proposes a flexible and generic paradigm over a wide variety of linguistic annotation types. FoLiA aims at practical usage, and the focus has been on the development of a rich infrastructure of tools to work with the format. Although many of the tools employed in the TTNWW project have adopted FoLiA either as input or output format, it should also be noted that other formats have been used as well — most notably the Alpino XML format for syntactic processing, but also other formats for more complex annotation structures. This emphasises the need for more convergence amongst these formats. In this respect FoLiA aims to provide a single comprehensive solution supporting a multitude of annotation types, and its ongoing development offers the possibility to extend it towards any annotation layers not provided yet. Such extensions can be informed by similar initiatives in this area such as the German Text Corpus Format (TCF) or the NLP Annotation Format (NAF); these may also provide alternatives in their own right, and the availability of good converters is therefore desirable for projects such as TTNWW. On a more practical level, interoperability should also address more ordinary issues, such as common tokenisation methods, to provide the opportunity to truly interrelate different annotation layers.

For linguistic enrichment to be effective in the web services/workflow paradigm, most already existing command-line tools had to be transformed to web services. In fact, the road towards this had already been paved in the prior CLARIN-NL (Odijk, 2010) demonstrator project TICCLops (Reynaert, 2014b), which not only turned an existing spelling correction system into a web application and service, but in fact delivered a generic solution for turning linguistic applications with a command-line interface into web applications and RESTful services.

The generic solution to turning any linguistic application into a web application/service, the Computational Linguistics Application Mediator, or CLAM (van Gompel, 2014; van Gompel and Reynaert, 2014),[3] was readily adopted by the TTNWW consortium to prepare their own linguistic applications for integration into the TTNWW workflow.

7.2 Text

7.2.1 Text Preprocessing

As a primary input TTNWW accepts digital texts that are either 'born digital' or the result of a digitisation process. To reduce the amount of Optical Character Recognition (OCR) noise in digitised texts TTNWW offers a corpus clean-up tool. The spelling and OCR post-correction system Text-Induced Corpus Clean-up (TICCL) was turned into the 'online processing system' TICCLops.[4] The approach is based on anagram hashing, which was first fully described and evaluated on English and Dutch in Reynaert (2005). In Reynaert (2010) it was applied to OCR post-correction of large

[3] Also available as Open Source software (via https://proycon.github.io/clam/).
[4] http://www.clarin.nl/node/70#TICCLops.

corpora. Two efficient modi operandi for obtaining the same end result, i.e. the set of vocabulary neighbours differing up to a specified number of characters, were presented. In a naive implementation based only on edit or Levenshtein distance (LD), each and every item in the vocabulary has to be compared to every other item. Anagram hashing typically reduces the number of comparisons required by several orders of magnitude, depending on the size of the vocabulary involved. Automatic correction of the Early Dutch Books Online corpus, which has a vocabulary of nearly 20 million items, is described in Reynaert (2014a).

7.2.2 Linguistic and Semantic Layers in TTNWW

To understand a text, key information can be inferred from the linguistic structure apparent in and across the sentences of the text. To determine who does what, to whom, when, where, why, and how, it is vital that the syntactic roles of words and word groups be identified, that entities be properly detected, and that different references to the same entities be linked.

TTNWW offers a number of tools that automatically identify this information. Of the following tools, tools 1 to 3 were developed and integrated into Frog, an Open Source natural language processing toolkit for the Dutch language[5] (van den Bosch et al., 2007). Almost all tools were integrated into TTNWW through the web service shell software package CLAM. We briefly discuss the tools independently:

1. Part-of-speech tagging and lemmatisation: identifying the syntactic roles of individual wordforms (e.g. 'paints' in 'Rembrandt used lead white paints for flesh tones' is a plural noun), and linking these wordforms to their standard dictionary lemma ('paint, noun'). The particular machine learning approach to part-of-speech tagging adopted for TTNWW, MBT (memory-based tagger), was originally introduced by Daelemans et al. (1996). Frog lemmatizes words and also performs a morphological analysis using a machine learning approach introduced in van den Bosch and Daelemans (1999).
2. Chunking: grouping words into syntactic phrases (e.g. 'lead white paints' and 'flesh tones' are noun phrases). Chunking can be used for different purposes, for example for identifying salient units in term extraction ('flesh tones' makes more sense as a term than 'flesh' or 'tones' individually) and for identifying the units for answering the 'who did what to whom…' questions ('Rembrandt' is the subject who 'used' 'lead white paints' as an object). The chunking approach in TTNWW, also based on the use of machine learning algorithms, was introduced by Daelemans et al. (1999). As training material, the Lassy Small Corpus was used, which is a syntactic treebank; tree structures from Lassy were converted into chunks with a rule-based script, and a memory-based tagger was trained on the chunked sentences.
3. Named entity recognition (NER): identifying proper names as names of people ('Rembrandt'), places, organisations, or other types of entities. For the system delivered for TTNWW, the developers experimented with a classifier ensemble in which a genetic algorithm was used for the weighted voting of the output of different classifiers (see Desmet and Hoste (2013) for more information). Since it performed equally well as the meta-learning approach, we opted for a single classifier based on the conditional random fields algorithm (Lafferty et al., 2001) as final NER classifier, which was delivered as a CLAM web service.
4. Coreference resolution: linking references to the same entities. For instance, if 'Rembrandt' is later referred to as 'He', the latter pronominal reference should be linked to Rembrandt

[5] Downloadable from http://languagemachines.github.io/frog/.

and not to any other entity mentioned in the text. For TTNWW, an existing mention-pair approach to coreference resolution (Hoste, 2005; de Clercq et al., 2011) which was further refined in the framework of the STEVIN projects COREA (Hendrickx et al., 2012) and SoNaR (Oostdijk et al., 2008; Schuurman et al., 2009; Oostdijk et al., 2012; Reynaert et al., 2012), was adapted to the pipeline of tools developed in the other work packages in TTNWW (e.g. the construction of markables was derived from Alpino output, cf. below). The resulting system was delivered as a CLAM web service.

5. Automated syntactic analysis is made available as a web service, by providing an interface to the Alpino parser for Dutch. Researchers can upload their texts to a web service which takes care of the required preprocessing, and takes care of running the Alpino parser. The result, syntactic dependency structures in the standard format developed in CGN (Schuurman et al., 2003), D-Coi (van Noord et al., 2006) and Lassy (van Noord et al., 2012), is made available to researchers in a simple XML format. Named entity recognition and classification, part-of-speech tagging and lemmatisation is integrated in the output of the parser.

   The underlying Alpino parser (van Noord, 2006; de Kok et al., 2011) is the de-facto standard syntactic parser for Dutch. It is a stochastic attribute value grammar where a hand-written grammar and lexicon for Dutch is coupled with a maximum entropy statistical disambiguation component. The parser is fairly accurate, with labeled dependency accuracy of around 90% on newspaper text. The speed of the parser varies with sentence length and ambiguity, but is about 2 seconds per sentence on average for typical newspaper text on standard hardware.

6. Spatiotemporal analysis: the STEx-tool (SpatioTemporal Expressions) for spatiotemporal analysis used in TTNWW enables researchers to deal with *incomplete* information and to analyze geospatial and temporal information the way the *intended* reader would have interpreted it, taking into account the relevant temporal and cultural information (using the metadata coming with the resource).

   Information presented in a text is never complete (Schuurman, 2007). What is meant is solvable by knowing where (and when) a text appeared originally. This information is stored in the metadata coming with a resource (Schuurman and Vandeghinste, 2010, 2011). In 'Hij doet opgravingen in het Turkse Sagalassos' (E: 'He is excavating in Sagalassos in Turkey'. De Morgen, 22-10-2011), 'Sagalassos' would be annotated as being situated in the Asian part of Turkey, where in 2011 the province of Antalya was located, Sagalassos having coordinates '37.678,30.519'. It was part of the region of Pisidia, and existed more or less from 10,000 BC until 600 AD. As input, STEx uses fully parsed sentences as provided by Alpino (cf. above).

7.2.3 Alignment

Alignment is a little bit of an outsider in the TTNWW project, as it is the only task involving another language than Dutch. Within the STEVIN project DPC (Dutch Parallel Corpus) an alignment tool chain was developed to arrive at a high-quality, sentence-aligned parallel corpus for the language pairs Dutch-English and Dutch-French, with Dutch as the central language (Paulussen et al., 2012). Within TTNWW this task included creating a web service for the alignment and the annotation of parallel texts (Dutch and English). The constraints of the alignment task involved a number of challenges not encountered elsewhere in TTNWW, due to the fact that more than one language is involved. The existing flow of the web service tool supposes the processing of just one input file (or a set of similar input files using the same processing chain), whereas an alignment task requires at least two input files. For the time being, the alignment service in TTNWW opts for a provisional solution.

7.2.4 Additions and Some Use Cases

Several other tools can be added, for example dealing with sentiment analysis, summarisation, semantic role labelling, information extraction, etc. TTNWW is designed to enable further extensions.

Some of the tools described above have been put to practice in large-scale follow-up projects. TICCL, for example, has been used as a standard preprocessing step in the Nederlab project (Brugman et al., 2016) for the Early Dutch Books Online corpus (Reynaert, 2016). Work in the Nederlab project also involves POS tagging using Frog to produce linguistically annotated corpora. Alpino is used in a broad range of projects; for HSS GrETEL (Augustinus et al., 2013), and Poly-GrETEL (Augustinus et al., 2016) are especially relevant, making it much easier to search in treebanks.

7.3 Speech

7.3.1 Tools Included in TTNWW

Speech recognition software provides HSS researchers with the possibility to transform audio signals into machine readable text formats. The speech recognition output could be reused as input for the text analysis processes, provided that the recognition rate is sufficiently high. Speech recognition systems are complex pieces of software requiring a fair amount of expertise to install and maintain. To make life easier for HSS users several web services were incorporated in TTNWW in which submitted audio files are automatically transcribed or where related tasks are performed. Several of these web services have been combined, resulting into ready-to-use workflows available to the HSS end user, see (Pelemans et al., 2012). Speech recognition web services are based on the SPRAAK software, see Demuynck et al. (2008).

1. Converter: extracts or converts speech files to the required .wav format for the Transcriber web service from a variety of other formats, including MP3 and video. This service is described in more detail in Pelemans et al. (2014).
2. Segmentation: within the TTNWW project, an audio segmentation tool was further improved and was made available via an easily accessible web service through CLAM. The provided audio segmentation tool first analyses the audio to find intervals which contain foreground speech without long interruptions, a process called speech/non-speech segmentation. Next, the speech intervals are divided into shorter segments uttered by a single speaker (speaker segmentation), and the speech fragments belonging to the same speaker are grouped (speaker clustering). These steps basically solve the "who-speaks-when" problem. Finally, the system identifies the language being spoken by each speaker (Dutch vs non-Dutch), enriches every audio fragment with extra non-verbal meta-information (e.g. is this music or telephone speech or dialect speech etc.), and detects the gender of every speaker. See Desplanques and Martens (2013), Desplanques et al. (2015), and Desplanques et al. (2014).
3. Diarisation: automatic speaker diarisation is the task of automatically determining: "who spoke when". On reception of an audio file, the web service labels each speaker in the recording ("SPK01", "SPK02", etc), it finds all speech segments and it assigns a speaker label to each segment. The result of the web service can be used as a preprocessing step in most state-of-the-art automatic speech recognition systems. The system is described in Hain et al. (2010) and Wooters and Huijbregts (2008).
4. Dutch Transcriber: uploads and transcribes Dutch broadcast news style of speech. Users have to answer some questions about the audio input so that the best recognition models are

chosen from a set of existing ones. More information on the transcription service may be found in Pelemans et al. (2014).

7.3.2 Additions and Some Use Cases

In addition to the services described above, several other useful speech services have been made available. Due to their experimental character they have not been incorporated into standard workflows for the TTNWW project. Some end users may however find some of them useful for their purposes. They are available as CLAM-enabled services and can be found on the www.spraak.org/webservice website. These include the

1. Dutch phoneme recogniser: this recogniser returns a phonetic transcription for the given audio input.
2. Grapheme to Phoneme Converter (g2p): this web service takes a list of (orthographic) Dutch words and returns a phonetic transcription for each of them.
3. Dutch speech and text aligner: takes as its input both an audio file and a text file and tries to align them. The output file contains the same words as the input, but with added timing information for every word. Optionally a speech segmentation file can also be given that contains speech/non-speech, male/female and speaker ID information as obtained from the speech segmenter described above.

These web services have already been put to use by several HSS users as demonstrated by some use cases:

- A test dataset of nine interviews from the KDC (Catholic Documentation Centre) at RU Nijmegen was prepared to be processed by the TTNWW speech chain. The interviews (total duration of 22.5 hours) were a small subset of the KomMissieMemoires series (KMM 873-880). All interviews obtained a CMDI metadata file which followed the OralHistoryDans profile (see https://catalog.clarin.eu/ds/ComponentRegistry) used in van den Heuvel et al. (2012).
- Currently, about 50 users have registered for the SPRAAK-based web services. Many users of the services want to check the potential performance of speech recognition on their specific task (often interview transcription and transcription of broadcast material) and find this a fast and flexible way to achieve this.
- Some applications and projects need existing tools, and instead of installing and maintaining these locally, prefer to call them over the web, as a RESTful service. One such example is the STON project (about semi-automated subtitling based on speech recognition), where a g2p converter is needed to provide phonetic transcriptions when new words are entered in the lexicon of the subtitling application, cf. Verwimp et al. (2016).

7.4 Web Services and Workflows

7.4.1 Web Service Delivery

All linguistic processing modules were required to be made available as web services. Web service deployment allows for a single service to be used by more non-technical users by lowering the barriers of installation and maintenance requirements. However, most modules had been constructed as command line tools as a result of previous projects. CLAM, (cf. Section 7.1.1), allows any command line tool to be wrapped as a web service — only parameters, input formats and output formats need to be described. Many of TTNWW's web services have been constructed in this manner. To facilitate transfer of web services from technology providers to CLARIN centres,

providers were requested to deliver services as fully installed virtual images. This reduces the installation overhead for CLARIN centres and ensures that web services are delivered according to the technology provider's recommended operating system. Images were deployed in an OpenNebula High Performance Cloud environment made available by SURFsara through a parallel project.

7.4.2 Combining Web Services in a Workflow

Depending upon the end user's requirements towards the desired linguistic annotation output, web services may need to be combined into pipelines. For example, to obtain coreference annotations the process entails tagging of textual input through Frog, followed by coreference annotation using the COREA service. To facilitate the full process, rather than just delivering an individual process, web services may be combined into workflows (Kemps-Snijders et al., 2012). In the CLARIN community two approaches were proposed for this. One approach allows end users to construct their own workflows by matching input/output requirements of individual services. Possible service combinations are determined using a generic chaining algorithm. This approach has been used in the WebLicht application (Hinrichs et al., 2010), created as part of the German CLARIN D-SPIN project. An alternative approach is to preconstruct complete workflows and provide these to the end user to perform a specific task. This has the advantage that end users can concentrate on task execution rather than task construction. Given the limited number of services and possible combinations for the available TTNWW services this approach was selected for this project. Incidentally, the WebLicht project now also offers predefined processing chains as an Easy Mode. For TTNWW, Taverna was selected as a workflow construction and execution framework. Upon selection of a specific task, the corresponding workflow definition is sent to a Taverna server monitoring execution and data transfer between contributing annotation services running in the HPC cloud environment. End users are shielded from workflow definitions, web services and execution environment through an easy-to-use user interface allowing them to upload their textual/audio data, to select the annotation task to perform and to collect the results afterwards.

7.5 Related Work

The web services and workflow paradigm has also been adopted by other projects to deliver processing services to the end user community. D-SPIN's WebLicht project mentioned before was an initiative of the German CLARIN community. The Danish CLARIN-DK project (Offersgaard et al., 2013) pursued a similar line with respect to automatic chaining of services into a workflow. The European PANACEA project (Poch et al., 2012), on the other hand, used the Taverna workbench and associated service registry to allow end users to construct and execute workflows in the NLP domain. Another recent service workflow is Galaxy, used by CLARINO[6] and LAPPS[7], amongst others.

7.6 Conclusions and Further Work

The TTNWW project delivers a suite of web services for the Dutch language domain. The CLAM software packaging software was broadly adopted by many teams to turn their shell-oriented software systems into web services. It has been demonstrated in the project that these services can be successfully combined into workflows. The resulting workflows are task-oriented in the sense that

[6] http://www.clarin.b.uib.no/about.
[7] http://www.lappsgrid.org/.

a series of web services are combined to deliver a specific end-user-oriented task. End-users only need to select a task and upload their resources, audio or text, after which execution and orchestration of the services is handled by the system. The TTNWW system is currently being revised as part of the ongoing CLARIAH project. Here, a new user workspace based on ownCloud[8] is expected to be added, as well as new features allowing the end user to search the resulting annotation files directly. As far as alignment, (cf. Section 7.2.3), is concerned, future work would involve to split up the original tasks into subtasks (i.e. cleaning, tokenisation and tagging) and to restrict the web service to its main task: i.e. alignment of parallel texts. In this way, the other web services can be used to handle the preparatory tasks, giving more flexibility in the development of tools and in administrating workflows. This will imply that all the other tasks require an extra language flag, so that language-specific modules can be used whenever necessary. Another aspect would consist in adapting the input format to the FoLiA format for input and output, so that the data format matches the requirements of the other tools in the web services chain.

References

L. Augustinus, V. Vandeghinste, I. Schuurman, and F. Van Eynde. 2013. Example-Based Treebank Querying with GrETEL - now also for Spoken Dutch. In *Proceedings of Workshop on Nordic language research infrastructure, NoDaLiDa 2013*, pages 423–428, Oslo, Norway. Linköping University Electronic Press.

L. Augustinus, V. Vandeghinste, and T. Vanallemeersch. 2016. Poly-GrETEL: Cross-Lingual Example-based Querying of Syntactic Constructions. In *Proceedings of LREC'16*, pages 3549–3554, Portorož, Slovenia. ELRA.

H. Brugman, M. Reynaert, N. van der Sijs, R. van Stipriaan, E. Tjong Kim Sang, and A. van den Bosch. 2016. Nederlab: Towards a Single Portal and Research Environment for Diachronic Dutch Text Corpora. In *Proceedings of LREC'16*, pages 1277–1281, Portorož, Slovenia. ELRA.

W. Daelemans, J. Zavrel, P. Berck, and S. Gillis. 1996. MBT: A memory-based part of speech tagger generator. In *Proceedings of 4th Workshop on Very Large Corpora, ACL SIGDAT*, pages 14–27, Copenhagen, Denmark.

W. Daelemans, S. Buchholz, and J. Veenstra. 1999. Memory-Based Shallow Parsing. In *Proceedings of CoNLL-99*, pages 53–60, Bergen, Norway.

O. de Clercq, V. Hoste, and I. Hendrickx. 2011. Cross-Domain Dutch Coreference Resolution. In *Proceedings of RANLP 2011*, pages 186–193, Hissar, Bulgaria.

D. de Kok, B. Plank, and G. van Noord. 2011. Reversible Stochastic Attribute-value Grammars. In *Proceedings of 49th Annual Meeting of ACL*, pages 194–199, Portland, Oregon.

K. Demuynck, J. Roelens, D. Van Compernolle, and P. Wambacq. 2008. SPRAAK : an open source SPeech Recognition and Automatic Annotation Kit. In *Proceedings of Interspeech 2008*, pages 495–498, Brisbane, Australia.

B. Desmet and V. Hoste. 2013. Fine-Grained Dutch Named Entity Recognition. *Language Resources and Evaluation*, 48(2):307–343. Springer.

B. Desplanques and J.-P. Martens. 2013. Model-based speech/non-speech segmentation of a heterogeneous multilingual TV broadcast collection. In *Proceedings of International Symposium on Intelligent Signal Processing and Communication Systems*, pages 55–60, Naha, Japan.

B. Desplanques, K. Demuynck, and J.-P. Martens. 2014. Robust language recognition via adaptive language factor extraction. In *Proceedings of Interspeech 2014*, pages 2160–2164, Singapore, Singapore.

[8] https://owncloud.org/.

B. Desplanques, K. Demuynck, and J.-P. Martens. 2015. Factor Analysis for Speaker Segmentation and Improved Speaker Diarization. In *Proceedings of Interspeech 2015*, pages 3081–3085, Dresden, Germany.

T. Hain, L. Burget, J. Dines, P.N. Garner, A. El Hannani, M. Huijbregts, M. Karafiat, M. Lincoln, and V. Wan. 2010. The AMIDA 2009 meeting transcription system. In *Proceedings of Interspeech 2010*, pages 358–361, Makuhari, Japan.

I. Hendrickx, G. Bouma, W. Daelemans, and V. Hoste. 2012. COREA: Coreference Resolution for Extracting Answers for Dutch. In P. Spyns and J. Odijk, editors, *Essential Speech and Language Technology for Dutch*, pages 13–126. Springer.

M. Hinrichs, Th. Zastrow, and E. Hinrichs. 2010. WebLicht: Web-based LRT Services in a Distributed eScience Infrastructure. In *Proceedings of LREC'10*, pages 489–493, Valletta, Malta. ELRA.

V. Hoste. 2005. *Optimization Issues in Machine Learning of Coreference Resolution*. Ph.D. thesis, University of Antwerp.

M. Kemps-Snijders, M. Brouwer, J.P. Kunst, and T. Visser. 2012. Dynamic web service deployment in a cloud environment. In *Proceedings of LREC'12*, pages 2941–2944, Istanbul, Turkey. ELRA.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of 18th International Conference on Machine Learning*, page 282–289, Williamstown, Massachusetts, USA. Morgan Kaufmann.

J. Odijk. 2010. The CLARIN-NL Project. In *Proceedings of LREC'10*, pages 48–53, Valletta, Malta. ELRA.

L. Offersgaard, B. Jongejan, and D. Haltrup Hansen. 2013. CLARIN-DK – status and challenges. In *Proceedings of Workshop on Nordic language research infrastructure, NoDaLiDa 2013*, pages 21–32, Oslo, Norway. Linköping University Electronic Press.

N. Oostdijk, M. Reynaert, P. Monachesi, G. Van Noord, R. Ordelman, I. Schuurman, and V. Vandeghinste. 2008. From D-Coi to SoNaR: a reference corpus for Dutch. In *Proceedings of LREC'08*, pages 1437–1444, Marrakech, Morocco. ELRA.

N. Oostdijk, M. Reynaert, V. Hoste, and I. Schuurman. 2012. The Construction of a 500-million-word Reference Corpus of Contemporary Written Dutch. In P. Spyns and J. Odijk, editors, *Essential Speech and Language Technology for Dutch*, pages 219–247. Springer.

H. Paulussen, L. Macken, W. Vandeweghe, and P. Desmet. 2012. Dutch Parallel Corpus: A balanced parallel corpus for Dutch-English and Dutch-French. In Peter Spyns and Jan Odijk, editors, *Essential Speech and Language Technology for Dutch*, pages 185–199. Springer.

J. Pelemans, K. Demuynck, and P. Wambacq. 2012. Dutch automatic speech recognition on the web: Towards a general purpose system. In *Proceedings of Interspeech 2012*, pages 9–13, Portland, Oregon, USA.

J. Pelemans, K. Demuynck, H. Van hamme, and P. Wambacq. 2014. Speech recognition web services for Dutch. In *Proceedings of LREC'14*, pages 3041–3044, Reykjavik, Iceland. ELRA.

M. Poch, A. Toral, O. Hamon, V. Quochi, and N. Bel. 2012. Towards a User-Friendly Platform for Building Language Resources based on Web Services. In *Proceedings of LREC'12*, pages 1156–1163, Istanbul, Turkey. ELRA.

M. Reynaert, I. Schuurman, V. Hoste, N. Oostdijk, and M. van Gompel. 2012. Beyond SoNaR: towards the facilitation of large corpus building efforts. In *Proceedings of LREC'12*, pages 2897–2904, Istanbul, Turkey. ELRA.

M. Reynaert. 2005. *Text-induced spelling correction*. Ph.D. thesis, Tilburg University.

M. Reynaert. 2010. Character confusion versus focus word-based correction of spelling and OCR variants in corpora. *International Journal on Document Analysis and Recognition*, 14:173–187. DOI: 10.1007/s10032-010-0133-5.

M. Reynaert. 2014a. Synergy of Nederlab and @PhilosTEI: diachronic and multilingual Text-Induced Corpus Clean-up. In *Proceedings of LREC'14*, pages 1224–1230, Reykjavik, Iceland. ELRA.

M. Reynaert. 2014b. TICCLops: Text-Induced Corpus Clean-up as online processing system. In *Proceedings of COLING 2014*, pages 52–56, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

M. Reynaert. 2016. OCR Post-Correction Evaluation of Early Dutch Books Online – Revisited. In *Proceedings of LREC'16*, Portorož, Slovenia. ELRA.

I. Schuurman and V. Vandeghinste. 2010. Cultural Aspects of Spatiotemporal Analysis in Multilingual Applications. In *Proceedings of LREC'10*, Valletta, Malta. ELRA.

I. Schuurman and V. Vandeghinste. 2011. Spatiotemporal annotation: interaction between standards and other formats. In *Proceedings of IEEE-ICSC Workshop on Semantic Annotation for Computational Linguistic Resources*, Palo Alto, California, USA.

I. Schuurman, M. Schouppe, T. Van der Wouden, and H. Hoekstra. 2003. CGN, an annotated corpus of Spoken Dutch. In *Proceedings of 4th International Workshop on Linguistically Interpreted Corpora, LINC-03*, pages 340–347, Budapest, Hungary.

I. Schuurman, V. Hoste, and P. Monachesi. 2009. Cultivating Trees: Adding Several Semantic Layers to the Lassy Treebank in SoNaR. In *Proceedings of 7th International Workshop on Treebanks and Linguistic Theories*, pages 135–146. LOT Occasional Series. Volume 12. Utrecht.

I. Schuurman. 2007. Which New York, which Monday? The role of background knowledge and intended audience in automatic disambiguation of spatiotemporal expressions. In *Selected Papers of the Seventeenth CLIN Meeting*, pages 191–206. Utrecht: LOT.

A. van den Bosch and W. Daelemans. 1999. Memory-based morphological analysis. In *Proceedings of 37th Annual Meeting of ACL*, page 285–292, San Francisco, California, USA. Morgan Kaufmann.

A. van den Bosch, G.J. Busser, S. Canisius, and W. Daelemans. 2007. An efficient memory-based morphosyntactic tagger and parser for Dutch. In *Selected Papers of the Seventeenth CLIN Meeting*, pages 191–206. Utrecht: LOT.

H. van den Heuvel, E. Sanders, R. Rutten, S. Scagliola, and P. Witkamp. 2012. An Oral History Annotation Tool for INTER-VIEWs. In *Proceedings of LREC'12*, pages 215–218, Istanbul, Turkey. ELRA.

M. van Gompel and M. Reynaert. 2013. FoLiA: A practical XML Format for Linguistic Annotation - a descriptive and comparative study. *Computational Linguistics in the Netherlands Journal*, 3.

M. van Gompel and M. Reynaert. 2014. CLAM: Quickly deploy NLP command-line tools on the web. In *Proceedings of COLING 2014*, pages 71–75, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

M. van Gompel. 2014. CLAM: Computational Linguistics Application Mediator. Technical report, Nijmegen: Radboud University. Technology Technical Report Series Report Number LST-14-02.

G. van Noord, I. Schuurman, and V. Vandeghinste. 2006. Syntactic Annotation of Large Corpora in STEVIN. In *Proceedings of LREC'06*, pages 1811–1814, Genoa, Italy.

G. van Noord, G. Bouma, F. van Eynde, D. de Kok, J. van der Linde, I. Schuurman, E. Tjong Kim Sang, and V. Vandeghinste. 2012. Large Scale Syntactic Annotation of Written Dutch: Lassy. In Peter Spyns and Jan Odijk, editors, *Essential Speech and Language Technology for Dutch*, pages 147–163. Springer.

G. van Noord. 2006. **A**t **L**ast **P**arsing **I**s **N**ow **O**perational. In *TALN 2006 Verbum Ex Machina, Actes De La 13e Conference sur Le Traitement Automatique des Langues naturelles*, pages 20–42, Leuven, Belgium.

L. Verwimp, B. Desplanques, K. Demuynck, J. Pelemans, M. Lycke, and P. Wambacq. 2016. STON: Efficient subtitling in Dutch using state-of-the-art tools. In *Proceedings of Interspeech 2016*, San Francisco, California, USA.

C. Wooters and M. Huijbregts. 2008. The ICSI RT07s speaker diarization system. In *Multimodal Technologies for Perception of Humans*, pages 509–519. Springer.