Daba: a model and tools for Manding corpora

Kirill Maslinsky
National Research University Higher School of Economics,
16 Soyuza Pechatnikov st., 190121 St.-Petersburg, Russia
kmaslinsky@hse.ru

Résumé. L'article traite du paquet des logiciels « Daba » créé dans le cadre du projet du développement des corpus pour les langues manding. Les particularités de ces langues ont motivé le développement des traits caractéristiques de ce logiciel. Le modèle de création du corpus a été, avant tout, testé sur le Corpus Bambara de Référence disponible en ligne en accès libre. La procédure de l'analyse morphologique et le schéma de l'étiquettage sont présentés en détail. Le Daba utilise le schéma de l'annotation morphologique inspiré par le glosage interlinéaire des exemples linguistiques. Une projection du modèle de présentation de l'information morphologique (sur la base du morphème) sur l'annotation traditionnelle de l'étiquettage (sur la base du mot) est prévue. Compte tenu du peu de standardisation de la forme écrite du bambara, le problème de la variabilité et sa présentation dans le corpus reçoivent une attention particulière.

Abstract. This article provides a brief overview of Daba software package created in the course of building corpora for Manding languages. Key software features are motivated by the tasks and problems characteristic of many African languages. The corpus-building model proposed here was initially developed for Bambara Reference Corpus which is available online and is freely accessible. The morphological analysis procedure and corpus annotation scheme are discussed in detail. Daba uses a morpheme-based morphological annotation scheme inspired by the interlinear glossed form of presentation of linguistic examples. A scheme mapping Daba's morpheme-based morphological information onto traditional word-based corpus annotation is provided. Since Bambara is characterized by a low level of written language standardization special attention is paid to the issues of representing variability in corpus annotation.

Mots-clés : TALN, analyseur morphologique, langues manding, bambara, annotation du corpus.

Keywords: NLP, morphological analyzer, Manding languages, Bambara, corpus annotation.

1 Introduction

In this article I discuss the design of corpora for Manding languages and describe the implementation of software tools developed for this purpose. Corpus building model proposed here was initially developed for the Bambara Reference Corpus (Vydrin, 2013), first published online in 2012[1]. Since then, the model has proved its usefulness in building a corpus of Guinean Maninka, a language closely related to Bambara. In addition, the model and tools are being tested in application to corpora of other minority languages not related to Manding, in particular to the Udihe language of the Tungusic family spoken in the Russian Far East. The Daba software suite is the core of this corpus solution.

Though corpus building is currently a well-established practice worldwide, it comes in many flavours. There may be numerous sources of diversity of corpora but I want to concentrate on those that shaped the corpus model proposed by the Bambara corpus working group and motivated the development of its software. The reasons lie partly in the sociolinguistic situation which characterizes patterns of language use and available linguistic resources, and partly in technical matters and available human resources.

Compared to most languages of Sub-Saharan Africa, linguistic resources available to the working group at the start of the project were good. A Bambara-French dictionary of over 10000 entries was kindly provided by Charles Bailleul in electronic form, and comprehensive dictionaries by Gerard Dumestre and Valentin Vydrin were also available (Bailleul, 2007; Dumestre, 2011; Vydrine, 1999). There's a long-standing tradition of linguistic research and teaching of Bambara

[1]The Bambara Reference Corpus is accessible at http://cormand.tge-adonis.fr/

in European and North American universities, and there are a number of grammatical descriptions. At the same time, texts in Bambara in electronic format are rather few and not easily accessible, so that much of the corpus building efforts were to be invested in digitizing printed material. Gerard Dumestre kindly provided us with the 100000-word collection of electronic texts in Bambara for the initial experiments. There were no linguistically annotated texts at our disposal.

The resources mentioned so far were sufficient to build a dictionary-based morphological analyzer for the automatic word-level annotation of Bambara texts. However, a number of issues arose from the low level of standardization characteristic of the Bambara written form (a situation typical of many other African languages, as well). Two official latin-based orthographic systems were used for Bambara in Mali; the new orthography has been in effect since 1987. Bambara is a tonal language, but both orthographic systems do not mark tones. As a consequence, each linguistic resource (dictionary or grammar) comes with its own tonal marking system developed by its author. There exist at least three such systems, by Charles Bailleul, Gerard Dumestre and Valentin Vydrin, not always allowing for simple conversion rules.

In addition, there are under-standardized areas in the existing orthographies, in particular the marking of word boundaries in composite words which are productively formed in Bambara and show considerable variation in writing. In any case, those orthographic rules that do exist are not always respected by native speakers of Bambara. More generally, Bambara is currently lacking a well-defined standard variety so that it is often unclear which of the competing lexical, spelling or tonal variants is to be considered standard, dialectal or idiosyncratic.

All these issues with lack of standardization show that variation in Bambara written form is a relevant feature for the sociolinguistic situation, and it should be taken into account in corpus building. A technical requirement for the corpus software is to retain all the information on variants and not to impose any subjective preferences for certain variants due to data unification procedures during corpus processing.

The idea of a Bambara corpus was coined by a group of three Russian linguists, specialists in Mande languages. I joined them as a linguist more proficient in IT. Our working group regarded the corpus mainly as a tool to serve the purposes of linguistic research and language teaching by non-native speakers. These aims shaped the general requirements for linguistic annotation in the corpus: a need for an annotation layer with consistent orthography and tonal marking, and a need for a layer of glosses — standardized equivalents in a European language. The obvious choice of a glossing language was French, widely used by researchers, learners and native speakers of Bambara.

Our group, composed of linguists involved in linguistic fieldwork and language documentation, regarded interlinear glossed text following Leipzig rules[2] as a suitable annotation model for Bambara texts. All group members were familiar with the Field Linguist's Toolbox software package[3] and used it as a reference point in discussion of annotation. It is no coincidence that most of the linguistic resources available to us, and most importantly Charles' Bailleul dictionary, were also in the Toolbox format. As a result, the design of the morphological annotation scheme is highly influenced by Toolbox, and Daba software supports the Toolbox format for dictionaries and annotated data. Compatibility with Toolbox made it easier for the group members to work on the content of linguistic resources, but resulted in a need for adaptation of our annotation to the model required by online corpus publishing software. These issues will be discussed in greater detail further in this article.

2 Bambara corpus software overview

The first major milestone for our working group was an online Bambara corpus of over 1 million words[4]. Following the example of "big" national language corpora, e.g. BNC or Russian National Corpus[5], the corpus was designed as consisting of two parts:

- Automatically annotated subcorpus with inevitable ambiguity due to grammatical homonymy. This is the larger part, currently over 1.4M words.
- Manually disambiguated subcorpus. This part is smaller, since disambiguation is a very labor-intensive task requiring a high level of competence in the language. Currently, the disambiguated Bambara subcorpus is over 0.23M words.

[2] See http://www.eva.mpg.de/lingua/resources/glossing-rules.php.
[3] See http://www-01.sil.org/computing/toolbox/.
[4] This milestone was reached in 2012.
[5] See http://ruscorpora.ru.

The corpus-building procedure established for the Bambara Reference Corpus can be split into several high-level tasks:

1. Automated morphological annotation of all texts.
2. Manual disambiguation of selected texts.
3. Adding metadata to all texts.
4. Creating an online search interface with flexible possibilities for concordance building.

We were unable to find a suitable software package or standalone tools that could be reused to solve tasks (1) through (3) for the Bambara corpus. The Daba[6] software package was developed by the author to fulfill these tasks. It is free and open source, written completely in the Python programming language and available at GitHub online code repository[7].

Tools for the first two tasks — automated morphological annotation optionally followed by manual disambiguation — were modelled on the process familiar to linguists: glossing of text in Toolbox. Basically, the morphological parser tries to split any wordform into a sequence of known morphemes using dictionary and morpheme combination constraints. If there are more than one possible parses, the user should interactively select the correct one (to disambiguate the word). In Toolbox, disambiguation is done at the same time as parsing and there's no way to leave an ambiguous parse result in place. Daba offers two separate utilities: a parser and a disambiguation tool. The parser processes text files non-interactively and saves the list of all possible parses for each token. The disambiguation tool allows a user to open the file produced by the parser and to select appropriate parse variants using a graphical user interface.

There exist two models of corpus metadata storage: either in a single database for all of the documents in a corpus or separately in each corpus document. Daba uses the second way, so that each corpus file contains all relevant metadata. The metadata editor provided in the Daba package allows a user to define the necessary metadata categories and to reuse metadata entries (e.g. an author's metadata) across several documents. More details on metadata annotation principles in the Bambara Reference Corpus are given in (Davydov, 2010).

NoSketchEngine was used as the online search interface and concordance building tool for the Bambara Reference Corpus. NoSketchEngine was selected for several reasons: it's free and open source; it is a mature project, alive and rather well supported; it supports — crucially — ambiguous values for annotation fields; and it provides a very flexible query language, CQL (Rychlỳ, 2007). NoSketchEngine is an open source variant of SketchEngine[8]. It is functionally limited compared to SketchEngine but it is more than sufficient for the purposes of the Manding corpora. The default NoSketchEngine code was modified to allow displaying the concordance in an interlinear-glossed style with annotation layers shown below the word form.

The Daba provides a separate utility to convert its internal data format into the vertical format required by NoSketchEngine to build corpus search indexes. This utility implements mapping of Daba's morphological annotation model onto annotation layers supported by NoSketchEngine. The algorithm of this mapping is further discussed in section 4.

3 Morphological analysis

Bambara, like all Manding languages, is a tonal language with a highly isolating morphology. This makes a traditional dictionary-based approach to morphological analysis a reasonable choice, since a simple dictionary lookup in a 10000-word dictionary covers about 90% of word forms in a Bambara text. There are several inflection affixes, some productive derivational affixes and a highly productive word composition process requiring heuristic rules in the morphological parser in addition to the dictionary.

The main downside of this simple approach to morphological analysis for Bambara is an abundant homonymy. The absence of tone-marking in the Bambara orthography, combined with the multiplicity of quasi-homonyms differing only in tone, leads to an ambiguity rate at about 70% in an automatically parsed text. This makes disambiguation a crucial task. Our group decided not to work on a sophisticated parser, keeping it simple and easy to implement, but instead to spend our efforts on manual disambiguation of text3s. This choice is in line with the argument by Sharoff and Nivre that

[6] *Dàba* is the Bambara word for 'hoe'. This name symbolizes the traditional and simplistic approach to morphological analysis used in the package. At the same time, it is an acronym for DAtabase for BAmbara.

[7] See http://github.com/maslinych/daba/.

[8] NoSketchEngine can be downloaded at http://nlp.fi.muni.cz/trac/noske.

linguists should invest in data annotation and not in rule sophistication (Sharoff & Nivre, 2011). Thus a gold standard morphologically annotated dataset could be produced sooner. This dataset can be further used for the development of statistical disambiguation tools.

Next in this section, morphological processing tools and resources are discussed in some detail.

3.1 Structure of morphological annotation

The model for morphological annotation used in Daba was inspired by interlinear morpheme-by-morpheme glosses as defined by Leipzig glossing rules. Each word form in a glossed text is annotated with a gloss and sometimes with a grammatical category (a part of speech tag). Any multimorphemic word is split into constituent morphemes with each morpheme having a corresponding gloss. An example of a glossed multimorpemic word from Bambara:

(1) báarabaliw
báara-bali-w
ptcp
travailler-PTCP.PRIV-PL

In Daba, morphological annotation of any word or morpheme is represented by a `Gloss` object. The gloss object is a triplet *word form — part-of-speech tag — gloss*. In the Daba interface it is conventionally written separated with colons:

(2) `báara:v:travailler`

For multimorphemic words this triplet is extended with a list of constituent morphemes, each being a Gloss object. A morphemes list is conventionally written in brackets after the basic triplet:

(3) `báarabaliw:ptcp: [báara:v:travailler bali::PTCP.PRIV w::PL]`

Note that affixes do not have a part-of-speech tag, and their glosses are standardized grammatical markers. Any field in a triplet can be left blank, e. g. the suprasegmental tonal article in Bambara is represented by Gloss object `::ART`.

The structure of a Gloss object is visualized in the Daba user interfaces in the form of a button labeled with form, part of speech, and gloss. Buttons for constituent morphemes are placed below the main word button:

báarabaliw (ptcp)		
báara (v)	bali	w
travailler	PTCP.PRIV	PL

The gloss object is recursive: since each morpheme is also represented by a Gloss object, it can have its own morphemes. This allows for flexible representation of the complex derivational structure of a word.

3.2 Lexical database: Bamadaba

A dictionary is the core component of morphological processing in Daba. Daba supports dictionaries in the Toolbox native format, also known as “standard format”. To be accepted and correctly processed as a lexical resource, a toolbox dictionary file should conform to a set of conventions. Each lexical entry is required to have a word form in the \lx field, a part-of-speech tag in the \ps field, and a gloss in the \ge field. For example:

(4) \lx báara
\ps v
\ge travailler

Optionally, phonetical, tonal, dialectal or other variants can be listed under lexical entry in several \va fields. When the dictionary is loaded into Daba, each lexical entry is transformed into a Gloss object used for analysis. For each variant, an

additional Gloss object is constructed that shares the part-of-speech tag and gloss with the main word form. All variants, including the main form, are treated as equivalent.

Other fields present in the source file and not used for constructing Glosses are simply ignored. Therefore, almost any Toolbox dictionary file developed for other purposes can be loaded into Daba with minimum modifications, mostly limited to field renaming.

A Bambara-French dictionary by Charles Bailleul (Bailleul, 2007), available in the Toolbox format, was used as a starting point for creating a lexical database for the Bambara corpus. Although the Toolbox format is transparently supported by Daba, much work was needed to transform a general-purpose dictionary into a lexical database suitable for morphological analysis[9]. Some issues were due to data inconsistencies mostly imperceptible by humans but intolerable for machine processing. For instance, there were over 150 different strings in the \ps field that should be normalized into a fixed set of part-of-speech tags. Also, a number of duplicate entries were present for variants already listed under some other lexical entry. Some other modifications of the dictionary were demanded by the goals of morphological annotation in the corpus. In particular, all French equivalents given in the original dictionary as translations were to be revised. For glossing purposes more semantically general single-word equivalents are strongly preferred to multi-word descriptive translations.

The resulting lexical database derived from the Bailleul's dictionary is supplied with dictionaries of proper names and is given its own name: Bamadaba. All dictionaries in Bamadaba are being continuously extended with new lexical entries found in the corpus.

3.3 Morphological parser

Daba's morphological parser uses two main resources: a lexical database and a grammatical file defining heuristic rules for processing word forms not found in a dictionary. The grammatical file is written using special syntax developed for Daba and consists of two parts: a list of *pattern rules* used for splitting a word form into constituent morphemes and processing instructions specifying the order of application of the rules. Technically, parser will work without a dictionary or grammar file, but its practical value will be limited in either case.

The parsing procedure for a single word form is organized as a sequence of pattern rule applications and dictionary lookups. When processing an input text, each word form is first of all transformed into a Gloss object with empty part-of-speech tag and gloss fields. A successful dictionary lookup returns a list of Gloss objects which represent possible interpretations of the word form. A successful pattern rule application splits a word form into morphemes. Grammatical morphemes are annotated by the parser itself and stems are looked up in the dictionary. In the processing instructions, a user specifies the order of pattern rule applications and dictionary lookups. (S)he can also define several points in the rule sequence where processing will be stopped if at least one fully glossed variant is present in the result list.

A pattern rule is twofold: it defines the rule applicability condition and the Gloss transformation operations. The first part specifies the context where rule is applicable by constraints on the input Gloss part-of-speech tag, word form and morpheme structure. Word form constraints are defined using regular expressions. The pattern rule usually also contains a morpheme splitting instruction also defined in terms of regular expressions. The second part of a pattern rule defines the annotation that should be assigned to the transformed Gloss and its morphemes.

An example of a simple pattern rule for Bamabara is a rule for analyzing privative participle forms ending in *-bali*:

(5) `pattern :v/ptcp: [ {|bali}:: ] | :ptcp: [ :v: :mrph:PTCP.PRIV]`

This rule states that it is applicable for forms having the part-of-speech tag "verb" or "participle" or an unspecified tag as well, ending in *-bali* (left part of the rule before the `|` symbol). The resulting form will be marked as a participle and split into two morphemes with the stem marked as verb and *-bali* marked as PTCP.PRIV (right part of the rule). The general form of the pattern rule and the use of regular expressions allows to annotate not only simple segmental morphemes (as in this example) but also morphemes with complex alternations depending on the phonetical context and suprasegmental morphemes.

A parser program processes an input file in the plain text format and produces parsed files in the native Daba format. Before the morphological analysis, the parser performs a number of auxiliary tasks: tokenizing input text, splitting sentences and normalizing orthography. The tokenizing and sentence splitting are done with a general built-in rule-based

[9]Cf. earlier attempt at transforming Bailleul's dictionary (1996 edition) into a digital lexical resource in (Enguehard *et al.*, 2012).

tokenizer. The orthographic normalization is performed for each token with a set of orthographic plugins, which are small independent python programs.

The orthographic conversion is not done as a separate step but is built into the parser because, according to the Bambara corpus methodology, all the variation in source forms should be retained. As a result of processing, the Daba parser saves the source word “as is” and annotates it with an orthographically normalized form and its morphological interpretation.

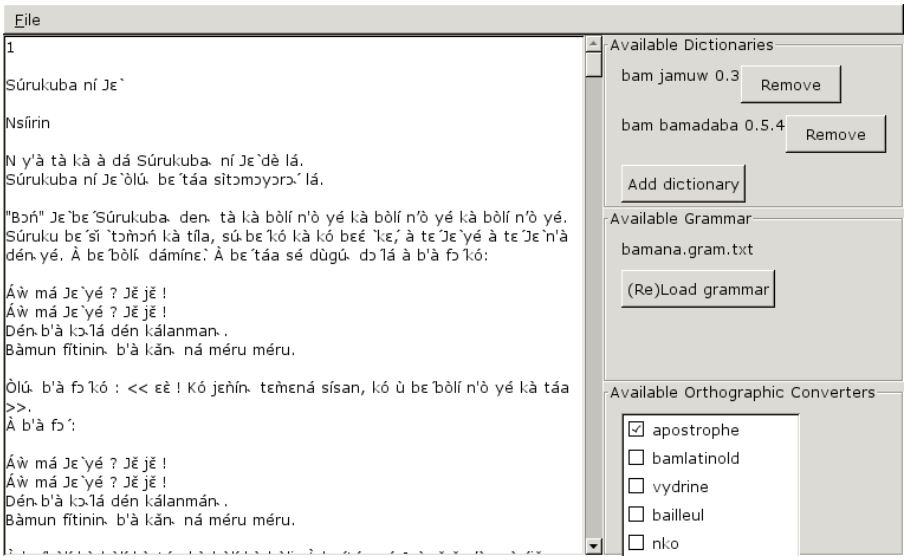

Figure 1: Graphical interface of the Daba morphological parser

3.4 Manual disambiguation

Files produced by the Daba parser can follow two routes: they are either directly converted into vertical format and uploaded into the disambiguated subcorpus, or passed to a Bambara-proficient operator for disambiguation. Daba provides a graphical user interface for the disambiguation which visualizes morphological annotation and lets a user choose correct annotation variants.

A sample screen showing a single sentence in the disambiguation interface is provided on fig. 2. The sentence is split into tokens, below each token there is a list of all possible morphological interpretations displayed as buttons. By pressing a button, a user can choose the correct variant. The resulting disambiguated file is saved in the same Daba format which can be further converted into the vertical format and uploaded into the disambiguated subcorpus.

4 Corpus annotation model

NoSketchEngine is used for the online publishing of Bambara Reference Corpus. NoSketchEngine is a general-purpose online corpus search interface with a feature-rich query language CQL and a possibility to store alternative (ambiguous) annotation variants for a token. In this section, the main procedures required to present morphologically annotated files in the Daba format as an online corpus are described.

The basic annotation unit in NoSketchEngine is a token (a word or a punctuation mark). An annotation for a token is presented in several pre-defined layers. The corpus administrator can define an arbitrary number of annotation layers containing linguistic data of any kind. Three very common attributes have special support in the web-interface. These are a *word* itself, its *lemma* (a normalized form used to aggregate tokens of the same word type), and a grammatical *tag*, usually the part of speech.

Annotated data should be loaded in NoSketchEngine in the so-called “vertical format”. This is a plain text file with one token on a line followed by a tab-separated list of its attributes. Each column in a vertical file corresponds to an annotation layer:

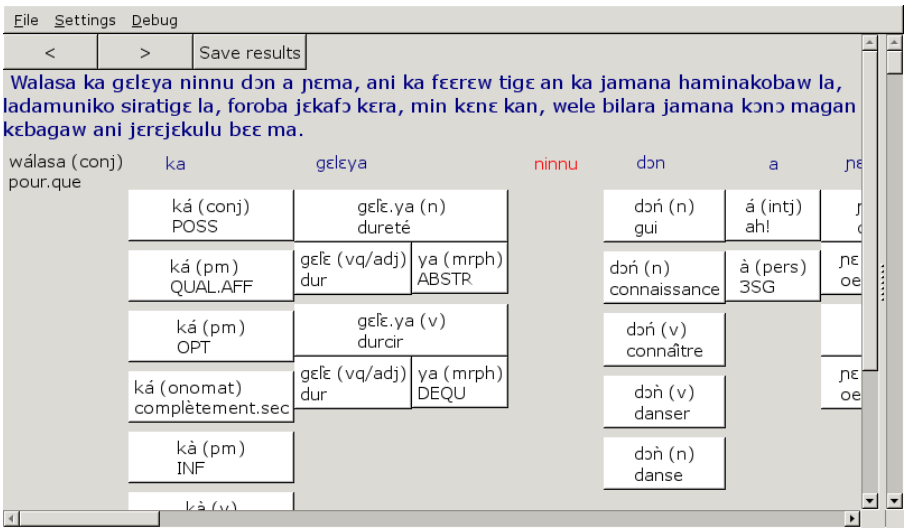

Figure 2: Graphical interface of the Daba disambiguation tool

(6) #word lemma tag gloss
baara báara n travail

The Daba's native morphological annotation model, described in section 3.1, doesn't have a direct analog of a lemma field for a word and also contains a morpheme-level morphological information that should not be lost in the process of converting Daba files into vertical format. A scheme mapping Daba's morphological annotation model onto wordform-based annotation for a NoSketchEngine solves these two primary tasks: lemma building and representation of morpheme-level grammatical information. A simple example of a token annotation following this mapping scheme is shown and commented below:

(7)

#word	lemma	tag	form	gloss	parts
baarabaliw	báara	ptcp|PL|PTCP.PRIV	báara-bali-w	travailler-PTCP.PRIV-PL	báara

The lemma for a word token is built from a Gloss object using a simple heuristic: the lemma is a concatenation of all constituent morphemes with the exception of those listed as inflectional. A closed short list of inflectional morphemes in Bambara makes this a simple and natural approach.

If a lemma has any variant listed in a lexical database, all these variants are added to the lemma attribute as alternatives. For example, the predicative marker *kà* has a graphical variant — the "contracted" form *k'*. For any *kà* and *k'* used in text, the lemma is identical and lists both variants: *kà|k'*. As a result, a search for the lemma *kà* or the lemma *k'* will return identical results with both contracted and full forms in the concordance. All kind of variants (phonetical, tonal, graphical, dialectal etc.) are treated this way. It makes all these kinds of variability systematically presented in the corpus and available for quantitative study.

The morpheme-based information contained in the Daba annotation scheme is translated into several different annotation layers in vertical format. The grammatical tag layer is filled with the part-of-speech tag for the whole word form and supplied with all standardized glosses (following Leipzig rules) of inflectional and derivational affixes constituting this word form. This way, search, e.g. for all plural forms becomes possible by a simple query for the grammatical tag *PL*.

For representing morphemic composition of a word form in a format most similar to interlinear glossing format, two annotation layers are added: *form* and *gloss*. The *form* attribute contains a hyphen-separated list of all morphemes of a token, and *gloss* contains a hyphen-separated list of glosses for the morphemes. These layers allow a corpus user to save pre-glossed examples from a corpus already represented in a conventional interlinear form.

The last attribute *parts* is filled with all stems (non-grammatical morphemes) found in a word form. This field is useful for an enhanced search in representing composite and derivative forms. A special search interface option called "Include composite and derivative forms" is implemented for the Bambara Reference Corpus using this field.

5 Discussion

The corpus-building procedure and the tools described in this article are highly influenced by the configuration of resources and limitations faced by the Bambara Corpus working group. Nevertheless, this procedure and tools can be regarded as a model for building corpora for other Manding languages since it provides solutions for the most demanding practical and methodological issues we needed to resolve. It has already proved its practical value for a Manding language closely related to Bambara, the Guinean Maninka. Work on the Bambara corpus lasted for several years before it reached a stage mature enough to publish the corpus online. Having the Daba suite ready, we were able to make the first online functional pre-release for the Maninka corpus roughly in a week starting from scratch. An application to other Manding and unrelated languages seems reasonable, too.

Each model has its own strengths and weaknesses. For Daba, the main deficiency probably lies in the simplistic implementation of morphological parsing. In the current implementation it is rather slow, taking couple of hours for processing a 1-million-word corpus. But for the practical purposes of the corpus, this is currently not an issue, since parsing any single text is done in a reasonable time. Parsing the full corpus is needed only at a corpus release moment, once in every several months. A more serious problem with the morphological analysis lies in the sequential parsing algorithm architecture which impedes the conditional application of pattern rules (apply a rule only if some other rule was matched). However, this kind of deficiency can be fixed in future Daba releases. The software implementation of graphical user interfaces in Daba also has all sorts of shortcomings characteristic of amateur programming with limited resources.

On the strong side of the Daba approach, an integration with the Toolbox data formats can be mentioned. This allows linguists to work with lexical resources in a familiar environment and saves an additional dictionary conversion step. In general, corpus building procedures of the Bambara Reference Corpus were designed to minimize the need for development of specific software for a corpus. A software developer is still a rare and expensive resource in a linguistic project. Freely available software is used to accomplish all corpus tasks where possible, notably, the NoSketchEngine for the complex software tasks of corpus indexing and querying and building a concordance web-interface. And yet, as the Daba software package proves by its mere existence, the task to avoid software development for a corpus can be achieved only to a certain degree.

6 Conclusion

Work on the Bambara Reference Corpus has shown that corpus building for a language with a low level of literacy and language standardization is different from the corpora of languages with long-established standards for orthography, lexical choice, and linguistic description. Bambara is a vivid example of such a low-standardized language. As research on such languages is usually done by linguists who are not native speakers and primarily address their work to the international linguistic community, the corpus model should meet the requirements of the tradition of data annotation and presentation in this community. Also, the lack of standardization should be regarded as a subject to study, not a subject for imposing an arbitrary standard by corpus architects. As a consequence, the most original parts of the corpus building model presented here are related to the integration of the interlinear glossed format into corpus annotation and a systematic representation of the lexical variation in the lexical database and in the corpus.

Future work on the Bambara Reference Corpus, as well as other Manding corpora, besides the obvious need for enlarging the corpus volume, should include a work on statistical algorithms aimed at reducing the ambiguity rate in an automatically parsed text.

Acknowledgments

The work on the Bambara Reference Corpus was supported by the RFBR grant #10-06-00219 “Development of the model for Manding electronic corpora (Maninka, Bambara)”.

This work is part of the program Investissements d’Avenir, overseen by the French National Research Agency, ANR-10-LABX-0083, (Labex EFL).

I am very grateful to Jean Jacques Méric for the constant flow of bug reports and feature suggestions for the Daba software package.

References

BAILLEUL C. (2007). *Dictionnaire Bambara-Français*. Bamako: Donniya, 3e édition corrigée edition.

DAVYDOV A. (2010). Towards the manding corpus: Texts selection principles and metatext markup. In *Proceedings of the Second Workshop on African Language Technology AfLaT*, p. 59–62.

DUMESTRE G. (2011). *Dictionnaire bambara-français suivi d'un index abrégé français-bambara*. Paris: Karthala.

ENGUEHARD C., KANE S., MANGEOT M., MODI I. & SANOGO M. L. (2012). Vers l'informatisation de quelques langues d'afrique de l'ouest (towards the computerization of some west-african languages) [in french]. In *JEP-TALN-RECITAL 2012, Workshop TALAf 2012: Traitement Automatique des Langues Africaines (TALAf 2012: African Language Processing)*, p. 27–40, Grenoble, France: ATALA/AFCP.

RYCHLÝ P. (2007). Manatee/bonito–a modular corpus manager. In *1st Workshop on Recent Advances in Slavonic Natural Language Processing*, p. 65–70: within MU: Faculty of Informatics Further information.

SHAROFF S. & NIVRE J. (2011). The proper place of men and machines in language technology. processing russian without any linguistic knowledge. *Komputernaja lingvistika i intellektual'nye tekhnologii: Po materialam Mezhdunarodnoj konferencii "Dialog"(Bekasovo, 25-29 maja 2011)*, p. 591–604.

VYDRIN V. (2013). Bamana reference corpus (brc). *Procedia-Social and Behavioral Sciences*, **95**, 75–80.

VYDRINE V. (1999). *Manding-English Dictionary (Maninka, Bamana)*, volume 1. St. Petersburg: Dimitry Bulanin Publishing House.