Technical Reports in Language Technology
Report No 2, 2008
http://www.njas.helsinki.fi/salama

SALAMA Dictionary Compiler - A Method for Corpus-Based Dictionary Compilation

Arvi Hurskainen
Institute for Asian and African Studies, Box 59
FIN-00014 University of Helsinki, Finland
arvi.hurskainen@helsinki.fi

Abstract

This paper describes a new approach to dictionary compilation, where a text corpus is converted into dictionary using a set of sophisticated language processing tools. The corpus-based dictionary compilation has well known advantages and significant advances have been made in it recently. However, various problems in implementation have restricted its wider application. The paper describes how a comprehensive language analysis system can be used for automating several labour-intensive phases in dictionary compilation. SALAMA Dictionary Compiler (SALAMA-DC) is based on a comprehensive language analysis system of Swahili. It produces (a) single-word headwords, (b) multiword headwords, (c) various types of cross-references, and (d) a user-defined selection of use examples in context, including controlled random selection, and selection based on frequent contexts. Particularly noteworthy is that the example texts can be translated to English and attached to each example. It should be noted that the whole process from raw text to dictionary takes place without manual editing in between. The system was tested with three text corpora.

Key Words: dictionary compilation, multiword expressions, rule-based machine translation, cross-reference, Swahili language, usage example

Abbreviations in text

ADC	automatic dictionary compilation
CG-2	Constraint Grammar, Version 2
MT	machine translation
MWE	multiword expression
SALAMA	Swahili Language Manager
SALAMA-DC	Swahili Language Manager adapted to dictionary compilation

Key to notation used in dictionary entries

{abudia}	headword as a lexical entry
(abudu)	base form of verb
{ worship }	gloss in English
[abudia]	key-word in use examples
V	verb

N	noun
IDIOM-V	idiom with verb as a member
9/10	noun class affiliation, class 9 means singular, class 10 means plural
AR	Arabic origin (etymological information)
APPL	applicative (type of verb extension)
627	frequency number
<ALA>	code indicating the source of the example text

1 Introduction

Dictionary compilation has traditionally been very labour-intensive, involving several man-years for producing even a moderate size dictionary. Therefore, it is understandable that computer has been taken to assist in various phases of the work. Constructing lexical databases for near-automatic dictionary compilation is one of the early solutions for automating the process (Sjögren 1988). In early years also the memory restrictions of computers placed obstacles for development. But even with powerful computers it has been difficult to computerize the whole process so that the result is satisfactory. There have been attempts to convert a corpus text into 'raw' dictionary entries, which then are manually corrected and further processed. Also use examples in context have been retrieved and selected from the corpus and then manually edited. However, the current state of affairs is not very encouraging, as Teubert (2004: 16) states, 'Even though most recent larger monolingual dictionaries pride themselves on having been informed by corpora, there still exist very few dictionaries which can claim that they are truly corpus-based'. If this is the case with monolingual dictionaries, how much worse might the situation be with bilingual dictionaries? And if a dictionary is corpus-based, it does not tell yet how fully it exploits the corpus.

Although the use of computational methods in dictionary compilation would speed up the compilation process tremendously (Storjohann 2006: 83), surprisingly few are efforts for automating the process. One of the early successful implementations was the production of annotated headwords with frequency information. The slow development process is perhaps due to the insufficient accuracy of the parsing systems and their poor suitability for dictionary compilation. The structure of comprehensive dictionaries is also so complicated and fine-tuned that the automation of the process does not first come to mind. Large structured databases, such as the WordNet databases for several languages, initiated at Princeton University (Miller 1995; Fellbaum 1996, 1998), and especially FrameNet (Fillmore et al. 2003; Baker et al. 2003; Atkins et al. 2003), help in dictionary compilation, but these do not aim at automatic dictionary compilation. This has led to the need to construct annotated tree-banks and word-nets, which are constructed either manually or with the help of annotation tools (Taylor 2003; Sampson 2003). Such problems include the identification and handling of various types of idioms and other multiword expressions (MWE), handling of homonyms, production of cross-references, inclusion of frequency information, and especially the inclusion of use examples in context. In this paper we shall address these problems and demonstrate how the above problems have been solved in SALAMA Dictionary Compiler (SALAMA-DC).

SALAMA-DC is a by-product of Swahili Language Manager (SALAMA), an environment for processing Swahili language computationally. SALAMA includes a comprehensive language analyzer of Swahili text, including morphological analysis, morphological and semantic disambiguation, syntactic analysis, and a bilingual translation module from Swahili to English. It has properties that make it suitable also for dictionary compilation. Particularly noteworthy is that, because it contains all modules needed for language analysis, it can take raw text as input. It is not restricted to a certain type of text - any text genre can be handled. Also domain-specific corpora can be processed to domain-specific dictionaries by making use of domain information included in the system.

Although the reliable production of dictionary headwords with necessary linguistic information is a challenging task, there are other tasks that are even more difficult to implement. These include the production of cross-references and the automatic management of representative use examples in context. For the latter operation to succeed satisfactorily, a number of requirements must be fulfilled.

(a) In heavily inflecting languages such as Swahili, the examples must be identified on the basis of the stem form, and not on the basis of the form that it has in text.

(b) It must be possible to define the maximum number of examples for each headword.

(c) It should be possible to guarantee that examples would be retrieved from each type of source text.

(d) Because in the corpus the number of examples is often big, it must be ensured that representative examples will be retrieved.

(e) Care must be taken that also rare words will get examples of use if present in the corpus.

(f) Multiword expressions must get examples of use.

(g) If a headword is a homonym, each member of the homonym must get examples of use, if found in the corpus. Also it is required that the examples are placed in appropriate places immediately after each member of the homonym.

(h) It must be possible to define the length of context in linguistically motivated ways, not only by defining the number of digits or words before and/or after the key-word. All these features have been implemented in SALAMA-DC. These points are elaborated on in some detail below, although full description is not possible in this article.

The task described above perhaps sounds over-ambitious, since language is, as Halliday (2005: 243) puts it '... perhaps the single most complex phenomenon in nature'. No claim is made here that the result achieved with computer processing is perfect and without need of checking and manual editing. However, significant parts of previously labour-intensive parts of the work can now be made automatically without losing the benefits of manual work.

2 Features of SALAMA and requirements of dictionary compilation

SALAMA is a system for making explicit the linguistic information available implicitly in text. The text written for a human reader must be re-written into a format that is ideal for computerized manipulation. To achieve this, we need to analyze the text morphologically, perform morphological and semantic disambiguation, isolate and

describe various types of multiword expressions, provide the words and expressions with definitions or glosses in another language, and perform syntactic analysis. The analyzed form of text can be manipulated in several ways, including the compilation of dictionaries. It depends on the coverage and accuracy of the language management system how well it suits for tasks such as dictionary compilation.

SALAMA-DC is a comprehensive system for producing dictionary entries from any word-form in Swahili. Therefore, it does not matter what type of text and how extensive the source corpus is. The system gives dictionary entries only for the words that appear in the corpus, including frequency information. Words that are not recognized by the system, such as misspellings and rare proper names, will be marked and separated for eventual manual handling.

There are two separate tasks in compilation based on SALAMA-DC. The first task is to produce the dictionary entries with appropriate linguistic information, and the second one is to find use examples for the headwords.

Some readers might like to know something about the architecture and technical solutions of SALAMA. Its fundamental basis is the linguistic theory, and the processing is guided by rules written on the basis of the theory. Statistical approaches are applied only when linguistic rules cannot be written. The morphological parser has two versions. The earlier parser, the construction of which started in 1985, was based on finite state methods and two-level description. Recently I have developed a two-phase method for morphological description. In it, regular expressions are used for performing the first level description. This meta-level description is then further processed into overt description with all required information displayed, including glosses in English. Morphological and semantic disambiguation as well as syntactic mapping is performed with Constraint Grammar Parser CG-2 (Tapanainen 1996). Also multiword expressions are described using this parser. Currently SALAMA operates in Linux environment, and it makes use of its many utilities and features.

2.1 Producing dictionary entries

In dictionary compilation based on corpora, a major issue is the identification of the head word form of each surface form of the word, or of a cluster of words in the case of MWEs. This is particularly challenging in languages with left-expanded inflection structures (Gouws and Prinsloo, 2005: 32). Advances in corpus linguistics have made us aware that although we can analyze language morphologically and syntactically, there is no guarantee that the meaning of the expression will be revealed as well (Siclair 2004). How can we produce accurate dictionary entries with appropriate meaning descriptions?

The analysis module of SALAMA produces text which is annotated in many ways, including glosses in English. Because only part of this information is needed in a dictionary, each analyzed word needs to be pruned so that only that information is left which is needed in the dictionary. Appropriate pruning is necessary also for getting the correct frequencies of each headword, because deviations in the form of lines cause the lines to be different and to be calculated separately.

There are many opinions on what constitutes a headword in a dictionary. It is obvious that all such words in base form that occur in text in some form should be listed in the dictionary. What this 'dictionary form' is depends on the language. From the

computational point of view this is irrelevant, because any form of the word can be selected as 'dictionary form'. A good dictionary includes also information on how the word is used. This calls for use examples, and we will deal with this question later. But the word may also have very specific uses in combination with one or more other words. Idiomatic expressions are examples of such uses. In them, it is not possible to derive the meaning of the expression on the basis of what we know of the basic meaning of the individual words that constitute the expression (Charteris-Black 2003). The meaning and use of such constructions should be included in the dictionary.

2.1.1 ***Entries with single-word headword.*** Most of the dictionary entries of Swahili have a single word stem as a headword. By stem is here meant the uninflected form of the word. In case of verbs, a stem can be the base form itself or a verb derived from the base form of that verb. Swahili has such extensions as causative, stative, applicative, and reciprocal verbs. It is more or less a question of taste, whether productively derived verb stems should be considered as headwords or whether they should be subcategorized under the basic stem. Exceptions are such derived verbs that have gone through the lexicalization process and have a special meaning; they are obviously separate headwords.

SALAMA-DC has both options. In one option, only basic verb stems are considered as headwords. In another option, also derived verbs are treated as headwords. With the latter option it is possible to get more fine-tuned results. It also ensures that use examples will be retrieved for each derived verb found in the corpus.

2.1.2 ***Entries with multiword headword.*** The treatment of multiword expressions (MWE) is a major problem in language technology. While they together constitute a concept, they may inflect in several ways, and there may be other words intervening between them (Fellbaum 2006: 349; Kramer 2006: 379). They also contain such important information that needs to be described in a dictionary. It is important to include in the dictionary such units as idioms, because their meaning cannot be derived on the basis of the meaning of individual members of the idiom. Idioms, which are a sub-set of MWEs, may also be polysemous and vague in meaning (Hümmer and Stathi 2006).

For languages such as Swahili the isolation of idioms is a complex process, because an idiom may contain inflecting units. For example, if a verb is part of an idiom, it may have thousands of forms in text. In SALAMA, isolation of MWEs is carried out after morphological disambiguation. The rules for isolating MWEs make use of the base forms of words and the linguistic tags attached to them. As a result of the isolation process we have a multiword stem with appropriate meaning. These word clusters can be handled as separate headwords in dictionary compilation. Also the retrieval of use examples of MWEs can be performed together with the retrieval of use examples of other stems. In fact, the system makes no difference between single-word and multiword handling.

In addition to idioms, there are many other kinds of MWEs, such as noun constructions and adjectival expressions. Also proverbs can be isolated with SALAMA and retrieved as a separate section in the dictionary. At the time of writing this, SALAMA contained a total of 2,127 idioms, 2,173 proverbs, and 6,338 multiword expressions of other types.

Where should a multiword expression be located in dictionary? Whether it should be alphabetized according to its most important member (Korhonen 2003: 78-80) or

according to other criteria is subject of discussion in lexicography (Espinal 2005; Gouws 1996; Kühn 2003; Rovere 2003). One approach is to locate them according to the first member in the dictionary. But it is also important to have cross references from other members of the MWE to the place, where the expression is described. SALAMA-DC facilitates this, so that for each member of a MWE a separate entry can be created, and a reference is made to the appropriate location.

(1) **[afya]** see [bwana_afya] [enye_afya] [enye_nguvu_na_afya]
[akili] see [enye_akili] [fanya_akili] [rukwa_na_akili] [teka_akili] [tia_akili]

The examples in (1) show only references to the appropriate locations without further information. If needed, it is also possible to give all lexical information here, as shown in (2). This alternative takes a bit more space but saves from unnecessary browsing.

(2) **[afya]** see [bwana_afya] N 9/6 HUM MW-N { health officer } 10
[afya] see [enye_afya] ADJ MW> { bonny } 17
[afya] see [enye_nguvu_na_afya] ADJ MW>>> { hale } 1
[akili] see [enye_akili] ADJ MW> { clever, cute } 32
[akili] see [fanya_akili] V IDIOM-V { use thinking } 1
[akili] see [rukwa_na_akili] V PASS IDIOM-V { lose one's mind, be mentally ill } 2
[akili] see [teka_akili] V IDIOM-V { control completely } 1
[akili] see [tia_akili] V IDIOM-V { take note of } 1

2.1.3 Homonyms. As it was Homonyms constitute special case in that while the lexical form is the same the interpretation is different. Swahili, being a noun class language, has a fairly limited number of true homonyms. Yet they have to be handled as separate entries in dictionaries. SALAMA-DC has facilities to identify each member of a homonym and give it the appropriate frequency information. The identification is carried out largely in the disambiguation process, where various linguistic rules help in defining the meaning of each member of the homonym cluster. Below we shall elaborate more on the retrieval of use examples for homonyms. In (3) we describe the head word 'funza', which is a noun and a verb. In Corpus 1 it was found as a noun only once, but as a verb ten times.

(3) **{funza}** N 9/10 { sand flea, larva, maggot, grub } HUM 1
{funza} V (funza) { educate, teach good manners } 10

2.1.4. Polysemy. A major problem in dictionary compilation is the accurate representation of polysemy (Boas 2005: 447). Polysemy is here understood as a case where one lexical unit has more than one semantic interpretation. The variation of meanings is best illuminated through use examples, and therefore it would be utmost important to find representative use examples for each meaning of the polysemous word. Here we are dealing with one of the major problems of automatic dictionary compilation.

The production of various meanings for the polysemous head word is not a big problem. They can be listed as part of the head word entry as in (4).

(4) **{fedha}** N 9/10-SG { 1 money, 2 silver, 3 brass, 4 coin, 5 capital, 6 finances }

However, if we want to find use examples for each meaning (assuming each meaning is found in the corpus), we have to be able to disambiguate semantically each occurrence of the word in text and encode it uniquely. This can be done using a different kind of representation for (4), as shown in (5).

(5) **{fedha1}** N 9/10-SG { money }
{fedha2} N 9/10-SG { silver }
{fedha3} N 9/10-SG { brass }
{fedha4} N 9/10-SG { coin }
{fedha5} N 9/10-SG { capital }
{fedha6} N 9/10-SG { finances }

The representation in (5) allows use examples to be attached automatically after each member of the polysemous cluster. In a system such as SALAMA-DC, a substantial part of polysemous words can already be handled so that use examples will be found for each meaning of the polysemous head word. However, this facility is still far from complete, and advances in it will be made along with success in carrying out semantic disambiguation.

2.1.5. ***Cross-references.*** Good dictionaries try to help the user by providing cross-references to the places, where additional information can be found. For example, a cross-reference to synonyms is useful information. Also reference to etymological information may be useful. In the case of multiword expressions there is need to have cross-references from some of the other parts of the expression to the place where the expression is described (Siepmann 2006: 11), but not necessarily from all parts of the expression. For example, if a genitive particle or a copula is part of the expression, it hardly needs a reference to all such multiword expressions, where that particle is a member. Also other types of cross-references are found in dictionaries. However, there is no common practice in providing cross-references in dictionaries.

From the computational viewpoint the task is challenging. I see two possible approaches for producing cross-references automatically. In one solution, information on cross-references will be included in the morphological analyzer, and that information is used in constructing cross-references in the dictionary. Another solution, applied here, is to produce cross-references computationally on the basis of the information available in the lexical entries of the MWEs. For example, from the sub-parts of MWEs it is possible to construct cross-references to the places, where the expression is described. It is also possible to construct cross-references to synonyms and near-synonyms. This is a complex task, where use is made of the English glosses and linguistic information. Also this is implemented in the current system.

3 Finding examples of use

A good dictionary includes examples of use, especially for words, the use of which is not self-evident. Manually compiled dictionaries usually have too few examples, and very

little effort is put for considering the representativeness and coverage of the examples (Boas 2005: 446). Computational processing makes it possible to retrieve all kinds of examples. How to access use examples in corpus is a major problem, and it is not enough that we have an unlimited access to examples in corpus (Varantola 1994; de Schryver, Gilles-Maurice, 2003: 167). In large corpora there are often too many examples, and some intelligent selection methods should be used for finding the representative examples. Also they should be sufficiently long for displaying the use context, but not unnecessarily long. In removing short examples care should be taken that rare words get examples even if they are too short to fulfil the standards set for the length of examples. How can all this be achieved automatically?

The system for retrieving examples of use in context is a complex process and only main points can be described here.

(a) We produce a frequency list of stems in the corpus, on the basis of which we retrieve examples of use. In producing the list, we can remove such cases as numbers, pronouns, proper names, and misspelled words. On the basis of the remaining list we can further decide which stems will not need examples of use.

(b) On the basis of the stem list we prepare two programs for marking all occurrences of each stem in the corpus. One program contains rare words, for which all examples will be retrieved. Another program contains frequent words, for which too short examples will not be accepted. This division is for taking care that in removing examples, we do not accidentally remove perhaps the only example for a word. For avoiding any mistakes in marking, each stem in a sentence must first be marked with a unique code.

(c) Excessively long sentences will be shortened for making sure that all stems in the sentence will be uniquely marked. In theory, of course, a program for encoding all stems also in long sentences could be constructed, but this is hardly economical. A maximum of 40 marked stems per sentence can be considered an upper limit. Sentences longer than this can be cut into two parts. This can be done fairly safely by splitting the sentences first at clause boundaries, an if such boundaries are not found and the sentence is still too long, then a forced cut is applied.

(d) When each stem of the sentence is uniquely encoded, we copy the stem to the beginning of the line and remove all extra stems in that sentence. Therefore, if we have, for example, twenty stems marked in a sentence, we get twenty example sentences from this single sentence, one for each marked stem.

If all example sentences will be joined together, we easily get an uncomfortably huge file. There are various ways of handling the problem. By shortening example sentences before marking them we can reduce the size of the resulting file. Especially the very frequent words cause the expansion of the result, because each occurrence of a word adds one more example sentence. By excluding part of the most frequent stems from example retrieval we achieve a considerable reduction.

There is also a more convenient way of solving the size problem. Marking the stems and the random selection of examples can be done as a single process without intermediate output files, and as a result we have a moderate size file with all use examples in context we need. If the source corpus for compiling the dictionary contains tens of millions of words, this is in practice the only practical method.

3.1 Shortening contexts

When we have a sorted list of example sentences and we know where in the sentence the stem is, we can shorten examples that we consider unnecessary long. There are two useful strategies for doing this. First we look for clause boundaries, and if found, the part of sentence beyond the boundary will be removed. This concerns boundaries before and after the key-word. This cutting operation can be done already before marking the stems, as described above. If no boundary is found, we cut the sentence at a distance of n words from the key-word. We can also check that the first word to be included is a word and not a punctuation mark. With these fairly simple methods it is possible to cut down the size of examples without losing essential information.

3.2 Selecting examples

A file with several millions of example sentences is not convenient for manual browsing; it simply contains too much information. In order to make the dictionary usable we have to remove most of the examples and leave the 'best' ones. How can we do this computationally? For words with only a few occurrences in the corpus this is not a problem; we can simply retrieve them all. The more frequent the word is the more complicated the process of choosing becomes. We must, as the thematic part of Lexicographica (2000) phrases, retrieve 'examples of examples' (Dolezal 2000).

Two methods for selecting examples are described here, the random selection and the selection based on frequent contexts.

3.2.1. Controlled random selection. In random selection, we retrieve a user-defined number of examples in context. The selection can be restricted in various ways. The system can be forced to select the examples from different sources, for example from different newspapers. Also too short examples can be removed if longer ones are available. In (6) is an example of selection, where each subcategory of a corpus is checked, and if examples are found, the maximum of two will be randomly retrieved from each.

(6)
{andika} V (andika) { write } 627
[andika] <ALA> kuwa ni kusoma na kuandika [andika] na kwamba jitihada za mwanakisomo
[andika] <ALA> waliotapeliwa fedha zao kujitokeza kuandika [andika] taarifa zao ili waliohusika wafikishwe
[andika] <KIO> licha ya kujifunza kusoma na kuandika [andika] vilevile alijifunza katekisimu na baadaye
[andika] <KIO> lile la majaji walikuwa wakiandika [andika] kile alichokisema, kisha jaji
[andika] <NIP> kuondoa hisia potofu kuwa kuandika [andika] wosia ni kuashiria au kuharakisha
[andika] <NIP> waandishi wa habari nchini kuandika [andika] habari sahihi zinazohusu kujikinga na
[andika] <NUR> Kondo alihoji ni vipi aliyeandika [andika] kwenye baiskeli yake ”Yesu

[andika] <NUR> ya siku zile na kuyaandika [andika] wakati mwingine wakiandikiana barua na
[andika] <RAI> na uwezo wa vyombo kuandika [andika] habari hizo kama vile waandishi walikuwa
[andika] <RAI> ya wasiojua kusoma na kuandika [andika]
[andika] <UHU> walikuwa wanajua kusoma na kuandika [andika], chini ya mpango wa elimu
[andika] <UHU> wengi hawana utaratibu wa kuandika [andika] wosia wakati wa uhai wao

3.2.2. Selection based on frequent contexts. In selection based on frequent context, we first find out what are the contexts where the word frequently occurs. This is done by looking at the key-word from a 'window', where some context is seen on both sides of the key-word. The optimal window size is found by experimenting with various sizes. If the corpus is millions of words, a window with two words on both sides of the key-word gives a fairly good result. With a smaller corpus the window must be narrower, so that sufficient frequencies for selection can be found. It also often turns out that a stem has more than one frequent context.

On the basis of the frequency list of the narrow context, a list of frequent contexts can be defined. It depends on the size of the corpus where the threshold for selection should be set. For example, any context occurring with a stem at least three times in Corpus 1 can be considered frequent, while in Corpus 3 the threshold can be set to 25.

Then, the chosen context as key, longer contexts will be retrieved. Now having retrieved from the corpus all the instances containing any of the frequent contexts, we have a list of examples, which contain such important information that we want to include in the dictionary. Whether we want to include them all or whether we want to do further pruning, depends on the size of the corpus as well as on restrictions to the size of the final dictionary. The experiments show that with Corpus 1 all examples can be included, while Corpus 2 and 3 require further pruning.

The examples in (7) and (8) clarify the process.

Step 1. Keyword seen through a 'window':

(7) tangu yalipoanza [anza] machafuko ya

Step 2. The example-length string with key-word:

(8) [anza] <DWE> wa Israel wameuawa tangu yalipoanza [anza] machafuko ya Intifada.

4 Joining pieces of the dictionary together

In the process described above we have produced three files as output: (a) the file containing lexical entries, (b) the file containing randomly selected examples in context, and (c) the file containing examples with frequent context. These three files can be joined together and sorted. All pieces will be nicely located in appropriate places, so that the headword comes first, then the random examples in context, and finally the frequent

context examples. If needed, the order of examples can be changed so that frequent context examples come first. An extract of a compiled dictionary is in (9).

(9) **{anza}** V (anza) { begin, start } 4289
[anza] <ALA> Fresh ya nao walionekana kuanza [anza] kutafuta usafiri mwingine, baada ya
[anza] <ALA> kamati ya chama hicho akaanza [anza] kumrushia madongo mwenyekiti wake mbele ya
[anza] <DWE> na hofu yoyote na kuanza [anza] kwenda makazini kama kawaida hii
[anza] <DWE> serikali mpya, Ujerumani imeshaanza [anza] bila kuchelewa kuitolea nchi yake
[anza] <KIO> hapo alinyanyuka na kuanza [anza] kunipiga na kwendea kisu alichokuwa
[anza] <KIO> anaonyesha mkono huo na kuanza [anza] kulia) hapo alinikamata hadi
[anza] <MAJ> Acb ilianza [anza] kufanya shughuli zake hapa nchini
[anza] <MAJ> kulala, na mchezo ukaanzia [anza] hapo"...
[anza] <NIP> "africa" One kuanza [anza] safari nchini karibuni,
[anza] <NIP> cha mjini Arusha, kitaanza [anza] uzalishaji wa dawa za maji
[anza] <NUR> "mimi" naanza [anza] kwa msemo wa hapo juu ya kwamba
[anza] <NUR> Aidha, lilipitishwa azimio kuwa kuanzia
[anza] <RAI> "Sisi tunataka kuanzia [anza] hapo tujue kama mali inayotangazwa
[anza] <RAI> maalum, Naila Jiddawi kuanza [anza] kujadiliwa jana
[anza] <UHU> Adhabu ya kufungiwa inaanza [anza] Januari 20, 2003",
[anza] fr <DWE> Waisraeli wameuawa, tangu yalipoanza [anza] machafuko ya Wapalestina, mwishoni
[anza] fr <DWE> Waisraeli 71 wameuawa tangu yalipoanza [anza] machafuko ya Wapalestina, Intifada
[anza] fr <DWE> itikadi kali, tangu yalipoanza [anza] machafuko ya Wapalestina mwezi wa
[anza] fr <DWE> kabisa mjini Jerusalem tangu yalipoanza [anza] machafuko ya Wapalestina zaidi ya miezi
[anza] fr <DWE> miongoni mwa Wapalestina, tangu yalipoanza [anza] machafuko ya Wapalestina mwezi wa
[anza] fr <DWE> wa Israel wameuawa tangu yalipoanza [anza] machafuko ya Intifada.
[anza] fr <DWE> wa itikadi kali tangu yalipoanza [anza] machafuko ya Intifada.

Here we have a dictionary entry for the verb anza. It is not a derived verb and its base form is also anza. It means 'begin, start'. Next follow randomly retrieved examples, maximally two from each source, provided with translation in English. Because the translation was produced and placed there fully automatically using SALAMA, some manual editing is needed for getting more fluent English. After randomly retrieved examples come examples that occur frequently in a certain context. The context is here 'tangu yalipoanza [anza] machafuko ya' and all are extracted from Deutsche Welle. This shows that when the window is 2 + 2 words, it excludes interesting contexts that are frequent but do not show up in this window, if the minimum frequency is 6 occurrences. By narrowing the window or by allowing less frequent occurrences we would get more variation to context. We could also impose the maximum number of each context type

and then get more variation without the risk of the total number of examples becoming too big.

If there are only a few examples in the corpus, all of them will be retrieved (10).

(10) **{abudia}** V (abudu) { worship } APPL 4
[abudia] <ALA> yanafanywa kwenye nyumba za kuabudia [abudia].
[abudia] <BIB> Yerusalemu ni mahali patupasapo kuabudia [abudia].
[abudia] <NIP> miradi ya nyumba za kuabudia [abudia].
[abudia] <NUR> Kwani sisi tunaabudia [abudia] kiwanda au mameneja wake mpaka

5 Translation of examples

Because SALAMA is a comprehensive environment for language manipulation, including machine translation, the automatic translation of examples has also been included to the system. Each context example gets a translation in English. In examples below, the translation was produced automatically using SALAMA, and it was also automatically located into the correct place. Please note that the non-extended verb form, for example *anza* and applicative *anzia*, yield the same headword in this system, if not otherwise specified for semantic reasons. For space restrictions, only part of the examples is here.

(11) **{anza}** V (anza) { begin, start } 4289
[anza] <ALA> kamati ya chama hicho akaanza [anza] kumrushia madongo mwenyekiti wake mbele ya (the committee of this party began to throw dirt on its chairman in front of)
[anza] <DWE> na hofu yoyote na kuanza [anza] kwenda makazini kama kawaida (and any fear and to begin to go to work as usual)
[anza] <DWE> serikali mpya, Ujerumani imeshaanza [anza] bila kuchelewa kuitolea nchi yake (the new government, Germany has begun without being late to give to its country)
[anza] <KIO> hapo alinyanyuka na kuanza [anza] kunipiga na kwendea kisu alichokuwa (on that moment he/she rose up and began to hit me and to go the knife which he/she had)
[anza] <KIO> anaonyesha mkono huo na kuanza [anza] kulia. (he/she shows this hand and begins to cry.)
[anza] <MAJ> ACB ilianza [anza] kufanya shughuli zake hapa nchini (ACB began to do its activities here in the country)
[anza] <MAJ> kulala, na mchezo ukaanzia [anza] hapo"... (to sleep, and the play began on that moment"...)
[anza] <NIP> "Africa" One kuanza [anza] safari nchini karibuni, ("Africa" One to begin the journey in the country soon,)
[anza] <NIP> cha mjini Arusha, kitaanza [anza] uzalishaji wa dawa za maji (of Arusha, it will begin the production of fluid medicines)
[anza] <NUR> "mimi" naanza [anza] kwa msemo wa hapo juu ya kwamba ("I" begin with the locution above that)

[anza] <RAI> "Sisi tunataka kuanzia [anza] hapo tujue kama mali inayotangazwa ("We want to begin here so that we would know if the ostensible property)
[anza] <RAI> maalum, Naila Jiddawi kuanza [anza] kujadiliwa jana (special, Naila Jiddawi to begin to be discussed yesterday)
[anza] <UHU> Adhabu ya kufungiwa inaanza [anza] Januari 20, 2003", (The punishment of being jailed begins January 20, 2003",)

The translation is based on the piece of text in the example, and not on the whole sentence. Therefore, important information is often missing, and this affects the translation result. But even with such shortcomings the system gives a rough translation, which can be manually corrected. Using whole sentences as context would yield even better results.

SALAMA-DC can also treat multiword examples, isolate them, and give appropriate translations. This is exemplified in (12).

(12) **{piga}** V (piga) { hit, beat } 647
{piga_pasi} V IDIOM-V { iron clothes } 3
[piga_pasi] <ALA> hapo mtu asipike wala asipige [piga_asi] pasi" (here man should neither cook nor iron clothes")
[piga_asi] <NIP> mara baada ya marehemu kumaliza kupiga [piga_asi] pasi nguo zake kisha wakamweka chini ya (immediately after the late to finish to iron his clothes then they placed him/her under)
{piga_picha} V IDIOM-V { photograph } 40
[piga_picha] <ALA> Ikulu kunywa chai na kupiga [piga_picha] picha na Rais Mkapa (the State House to drink tea and to photograph with President Mkapa)
[piga_picha] <ALA> wa na pili, wapige [piga_picha] picha, alionekana kugoma (of and second, they should photograph, he/she was seen to boycott)
[piga_picha] <DWE> Au kumpiga [piga_picha] picha au hata kupeana naye (Or to photograph or even to give to each other with him/her)
[piga_picha] <DWE> kutoka Ujerumani, walijitahidi kupiga [piga_picha] picha za ukumbusho na kiongozi wao (from Germany, they made an effort to photograph the commemoration and their leader)
{piga_ramli} V IDIOM-V { divine } 4
[piga_ramli] <KIO> anakwenda kwa mganga ili kupiga [piga_ramli] ramli na kuongeza imani za ushirikina (he/she goes to the medical person in order to divine and to increase the faith of superstition)
[piga_ramli] <KIO> ikambidi amtume mtaalam wa kupiga [piga_ramli] ramli kuhusu nyota hiyo (he/she was obliged to send to him/her the expert of divining concerning this star)
[piga_ramli] <KIO> kwenda kwa mganga wa kupiga [piga_ramli] ramli, hujui kuwa imani ya (going to the medical person of divining, you do not know that the faith of)
[piga_ramli] <RAI> kuachana na mtindo wa kupiga [piga_ramli] ramli (to abandon the style of divining)

Homonyms need examples for each member of the group. An example of how this is produced by SALAMA-DC is in (13).

(13) **{jibu}** N 5/6 { answer, reply, response } AR
[jibu] <DWE> wanahoji wafuasi wake wanaohakikisha jibu [jibu] litakuwa kali (they question their followers who ascertain that the answer will be sharp)
[jibu] <KIO> Aliposikia jibu [jibu] hilo, mwuaji huyo akamuua (When he/she heard this answer, this killer killed him/her)
[jibu] <NUR> anashangazwa na madai ya majibu [jibu] ya Rais wa Jamhuri ya Muungano, Rais (he/she is amazed by the assertions of the answers of the President of The United Republic, the President)
[jibu] <NUR> Jibu [jibu] alilonitupia lingekuwa mshale saa hizi (The answer which he/she threw at me would be the arrow these hours)
{jibu} V (jibu) { answer, reply, respond, react to } AR 1009
[jibu] <ALA> kalamu na karatasi na kumjibu [jibu] Makamba kuwa yeye anampenda (the pen and the piece of paper and to answer Makamba that he likes him/her)
[jibu] <ALA> la polisi lakini hawakutakiwa kujibu [jibu] lolote kwenye kesi ya mauaji (of the police but they were not wanted to answer anything in the case of the murder)
[jibu] <DWE> Akijibu [jibu] hoja hiyo ya Papa John (When answering this argument of Pope John)
[jibu] <KIO> Akakaa tu, asijibu [jibu] lolote wala kusalimia (He/she sat only, he/she would not answer anything nor greet)
[jibu] <KIO> mjomba akadakiza nami nikajibu [jibu] kuwa ni kweli ndivyo ilivyo (the maternal uncle interrupted speech and I answered that it is true as it is)
[jibu] <MAJ> na serikali mara nyingi jibu [jibu] lao huwa chama hakina habari (and the government many times their answer is that the party does not have the news)
[jibu] <NIP> alisema Rais Mkapa wakati akijibu [jibu] swali (said President Mkapa the time when answering the question)
[jibu] fr <ALA> Akijibu [jibu] swali hilo, Waziri Mwapachu (When answering this question, Minister Mwapachu)
[jibu] fr <ALA> amefikishwa mahakamani hapo kujibu [jibu] shtaka la kuiba katuni tatu (he/she has been brought to the court here to answer the complaint of stealing three cartoons)
[jibu] fr <ALA> Bungeni leo wakati alipokuwa akijibu [jibu] swali la la Mhe (In the parliament today the time when he/she was answering the question of of Hon.)
[jibu] fr <ALA> amefikishwa katika mahakama hiyo kujibu [jibu] shtaka la wizi (he/she has been brought to this court to answer the complaint of theft)
[jibu] fr <ALA> ametoa hiyo jana alipokuwa akijibu [jibu] swali la Mhe (he/she has given this yesterday when he/she was answering the question of Hon.)

As said above, SALAMA-DC is able to find and translate also proverbs. In Corpus 2, the following proverbs were found:

(14) {Asiyefunzwa na mamaye, hufunzwa na ulimwengu} (Who is not taught by one's mother is taught by the world) 1
{Bendera inafuata upepo} (The flag follows the wind) 1
{Biashara haigombi} (Business does not argue / business is negotiable) 1
{Hujafa, hujaumbika} (You are not yet dead, so you are not yet fully created) 2

{Haba na haba hujaza kibaba} (Little and little fills up the tin) 1
{Hasira hasara} (Anger brings damage) 1
{Kidole kimoja hakivunji chawa} (One finger does not kill a louse) 1
{Kutangulia si kufika} (To go first is not to arrive) 2
{Kutoa ni moyo, usambe ni utajiri} (Giving is from heart, do not say it is from richness) 1
{Majuto ni mjukuu} (Regret is like grandchild) 2
{Mgeni njoo mwenyeji apone} (Let the guest come and the host be rescued) 1
{Mkono mtupu haulambwi} (The empty hand is not licked) 1
{Mla nawe hafi nawe ila mzaliwa nawe} (The one who eats with you does not die with you like the one who was born with you) 1
{Msafiri kafiri} (The traveller is an unbeliever) 1
{Bendera hufuata upepo} (The flag follows the wind) 3
{Penye wengi hapaharibiki neno} (Where there are many people, the word is not misused) 1
{Sheria ni msumeno} (The law is like a saw) 1
{Tamaa mbele mauti nyuma} (Greediness first, death later) 1

6 Tests with three corpora

SALAMA-DC was tested with three text corpora. Corpus 1 (148,700 words) contains five fiction books of E. Kezilahabi. Corpus 2 (3 million words) contains only news texts. Corpus 3 (20 million words) is a combination of all kinds of texts. However, no claim is made on its representativeness of Swahili language use.

Because SALAMA-DC includes computationally heavy processing, tests were needed for finding out possible problems in working memory. Tests were made with a laptop that had only 512 kb working memory. All phases in processing Corpus 1 and Corpus 2 succeeded without problems. The processing of Corpus 3 required splitting the program for retrieving all example sentences because of insufficient working memory. With a modern ordinary laptop with more powerful working memory this was not a problem.

Another problem to be tested was to see how the size of the corpus affects the need to handle use examples, so that the best possible results would be achieved in each case. Corpus 1 was small compared with the other two, and no restrictions for retrieving examples with frequent context seemed appropriate. Also in Corpus 2, no restriction of frequent contexts was applied. Corpus 3 required restriction of frequent context examples, some of them having hundreds of examples in the corpus. This was done with the same method as the retrieval of random examples. Maximally three examples were retrieved from each text source.

Table 1 summarizes the properties of the dictionaries compiled from Corpus 1, 2 and 3.

Table 1. Properties of test dictionaries compiled from three corpora.

	Corpus 1 148,700 words	Corpus 2 3 million words	Corpus 3 20 million words
Headwords	5,961	15,189	25,576

(single word)			
Headwords (multiword)	1,001	1,489	2,381
Cross references (multiword)	1,117	1,495	2,634
Cross references (synonyms)	2,085	2,945	5,212
Examples in context (randomly selected)	10,112	105,818	173,082
Examples in context (based on frequent contexts)	1,883	31,258	11,048

We see that the text written by one novel writer (Corpus 1) contains a rather limited vocabulary. Corpus 2, although fairly large, contains news texts only, and the vocabulary is quite limited in regard to corpus size. Corpus 3 contains mostly news text, but also many other genres are included. As a result, the number of headwords is fairly large. Examples with context are not comparable at all, because in each case different criteria were applied. In Corpus 1, two examples at maximum were retrieved for each headword, and all frequent context examples were included. Corpus 2 was divided into sections according to source text, and maximally two examples were retrieved from each source text. Therefore, the number of examples is high compared with Corpus 1. Also the number of frequent context examples is high, because no restriction was applied. Corpus 3 was divided into twenty one sub-categories, and the system was allowed to retrieve maximally two examples from each sub-category. As a consequence, the total number of randomly retrieved examples is high. Examples with frequent contexts were restricted so that maximally three examples from each sub-category were retrieved. As a result the number of retrieved examples is fairly limited.

7 User interfaces

SALAMA-DC produces a dictionary in text format, all as a single file. Some users prefer this format, because in it they can use their own favourite search and retrieval facilities. For less computer-literate users the file can be converted to various database structures, hyperlink structures (ten Hacken 2006: 249) and other formats, because the dictionary has a strict and systematic encoding of all elements. In future it is perhaps possible to use the system described here for producing linguistic resources of Word Net type, which in turn can be linked into larger multilingual networks of resources (Janssen 2004; Vossen 2004).

8 Further considerations

SALAMA Dictionary Compiler is a general system for corpus-based dictionary compilation. It takes any text, without previous encoding, as input and produces a

dictionary with headwords followed by selected use examples in context. Perhaps the biggest strength of the system is in its ability to combine the headword with use examples with translation. Except for keying in a few command calls, the user does not manually interfere with the process. Choices can be made in regard to words that do not need examples, the length of context, the maximum number of examples per headword, etc. The output can be tuned in many ways. Some may prefer to retrieve more examples and make the final choice manually. Others may prefer a near-final result with fewer examples.

Although the phases in dictionary compilation have been automated in the system, there are still critical points that can distort the quality of the result. The source corpus must be compiled carefully, because the system simply makes explicit what is in the corpus, and does not add information. Therefore, the corpus design has immediate effects on the contents of the dictionary as well as to the frequency counts (Mahlberg 2004; Goebel 2000). Another critical point is the language analyzer, which hardly ever is as good as it should be. Especially semantic disambiguation is difficult to implement reliably. But it is encouraging to realize that any improvements in the quality of the corpus and in the analysis system contribute positively to the compiled dictionary. It is also important to note that the automatically compiled dictionary needs manual checking and correction. However, the work needed is only a fraction of the work needed in manual compilation.

Because the system is not language-specific, it suits to dictionary compilation also in other languages. The major requirement is that there is a fairly good language analyzer available. The ability of the system to handle MWEs depends entirely on the analysis system. For compiling a bilingual dictionary, also an electronic bilingual vocabulary is needed for converting the lexical entry into the required form in the target language. Such tools are increasingly available for all major languages.

Above I have described only the main features in automatic dictionary compilation. What about if more words are needed than found in the corpus? Such words can be extracted from the analyzer lexicon itself, but no use examples will be found from the corpus. It is also possible to enlarge the source corpus, preferably with different types of texts for getting a better coverage. In doing this, the vocabulary extraction and example extraction should be run from the same corpus. If not, there is a danger that there will be headwords without examples, and examples without headwords.

It should be noted that the dictionary compilation using the above methods opens entirely new possibilities in regard to the size of the dictionary. In our test with Corpus 3, the original compiled dictionary with all examples of use was reduced to two percent of the original by using selection and reduction of examples. Yet the final dictionary was almost twenty times the size of the current Standard Swahili dictionaries. It is clear that there is not much point in printing such a dictionary. However, with a user-friendly interface it will be an excellent tool in computational environment.

References

Abeillé, A. (ed.), 2003. *Treebanks: Building and Using Parsed Corpora.* Dordrecht, Boston and London: Kluwer Academic Publishers.

Atkins, S., Rundell, M., and Sato, H. 2003.'The Contribution of FrameNet to Practical Lexicography.' *Lexicography* 16(3): 333-357.

Baker, C. F. Fillmore, C. J., and Cronin, B. 2003. 'The Structure of the FrameNet Database.' *Lexicography* 16(3): 281-296.

Boas, H. C. 2005. 'Semantic Frames as Interlingual Representations for Multilingual Lexical Databases.' I*nternational Journal of Lexicography* 18(4): 445-478.

Charteris-Black, J. 2003. 'A Prototype Based Approach to the Translation of Malay and English Idioms.' in S. Granger, J. Lerot, and S. Petch-Tyson (eds.), *Corpus-based Approaches to Contrastive Linguistics and Translation Studies.* Amsterdam and New York: Rodopi, 123-140.

De Schryver, G.-M. 2003. 'Lexicographer's Dreams in the Electronic-dictionary Age.' *International Journal of Lexicography* 16(2): 143-199.

Dolezal, F. F. M., 2000. 'From Text to Dictionary.' *Lexicographica* 16: 1-7.

Espinal, M. T. 2005. 'A Conceptual Dictionary of Catalan Idioms.' *Lexicography* 18(4): 509-540.

Fellbaum, C. 1996. 'WordNet: Ein semantisches Netz als Bedeutungstheorie.' in J. Grabowski, T. Herrmann, and G. Harras (eds.), *Bedeutung, Konzepte, Bedeutungskonzepte*. Opladen: Westdeutscher Verlag, 211-230.

Fellbaum, C. 1998. 'Towards a Representation of Idioms in WordNet.' in *Proceedings of the COLING/ACL Workshop on Usage of WordNet in Natural Language Processing Systems.* Montreal, CA, 52-57.

Fellbaum, C. 2006. 'Corpus-based Studies of German Idioms and Light Verbs.' *International Journal of Lexicography* 19(4): 349-360.

Fillmore, C. J., Johnson, C. R., and Petruck, M. R. L., 2003. 'Background to FrameNet.' *Lexicography* 16(3): 235-250.

Goebel, U. 2000. 'From Text to Example: The Early New High German Project.' *Lexicographica* 16: 25-46.

Gouws, R. H. 1996. 'Idioms and Collocations in Bilingual Dictionaries and Their Afrikaans Translation Equivalents.' *Lexicographica* 12: 54-88.

Gouws, R. H. and Prinsloo, D. J. 2005. 'Left-expanded Article Structures in Bantu with Special Reference to Isizulu and Sepedi.' *International Journal of Lexicography* 18(1): 25-46.

Halliday, M. A. K. 2005. 'Computing Meanings: Some Reflections on Past Experience and Present Prospects.' in J. J. Webster (ed.), *Computational and Quantitative Studies.* London and New York: Continuum, 239-267.

Hümmer, C. and Stathi, K. 2006. 'Polysemy and Vagueness in Idioms: A Corpus-based Analysis of Meaning.' *International Journal of Lexicography* 19(4): 361-377.

Janssen, M. 2004. 'Multilingual Lexical Databases, Lexical Gaps, and SIMULLDA.' *International Journal of Lexicography* 17(2): 137-160.

Korhonen, J. 2003. 'Phraseologismen in neuerer deutsch-finnischer Lexicografie.' *Lexicographica* 19: 73-96.
Kramer, U. 2006. 'Linguistic Lightbulb Moments: Zeugma in Idioms.' *International Journal of Lexicography* 19(4): 379-395.
Kühn, P. 2003. 'Phraseme im Lexicographie-Check: Erfassung und Beschreibung von Phrasemen im einsprachigen Lernerwörterbuch.' *Lexicographica* 19: 97-118.
Mahlberg, M. 2004. 'The Evidence: Corpus Design and the Words in a Dictionary.' *Lexicographica* 20: 114-129.
Miller, G. A. 1995. 'WordNet: A Lexical Database for English.' *Communications of the ACM* 38(11): 39-41.
Rovere, G. 2003. 'Phraseme in zweisprachigen Wörterbüchern mit Italienisch und Deutsch.' *Lexicographica* 19: 119-139.
Sampson, G. 2003. 'Thoughts on Two Decades of Drawing Trees.' in A. Abeillé (ed.), 23-41.
Siepmann, D. 2006. 'Collocation, Colligation and Encoding Dictionaries.' *International Journal of Lexicography* 19(1): 1-39.
Sinclair, J. 2004. 'Meaning in the Framework of Corpus Linguistics.' *Lexicographica* 20: 20-32.
Sjögren, C. 1988. 'Creating a Dictionary from a Lexical Database.' in M. Gellerstam (ed.), *Studies in Computer-Aided Lexicology.* Stockholm: Almqvist & Wiksell, 299-338.
Storjohann, P. 2006. 'Korpora als Schlüssel zur lexikografischen Überarbeitung – der neue Dornseiff.' *Lexicographica* 21: 83-96.
Tapanainen, P. 1996. *The Constraint Grammar Parser CG-2.* Publications No. 27. Department of General Linguistics, University of Helsinki.
Taylor, A. 2003. 'The Penn Treebank: An Overview.' in A. Abeillé (ed.), 5-22.
Ten Hacken, P. 2006. 'Word Formation in an Electronic Learners' Dictionary: ELDIT.' *International Journal of Lexicography* 19(4): 243-256.
Teubert, W. 2004. 'The Corpus Approach to Lexicography.' *Lexicographica* 20: 1-19.
Varantola, K. 1994. 'The Dictionary User as Decision Maker.' in: W. Martin, W. J. Meijs, M. Moerland, E. Ten Pas, P. G. J. Van Sterkenburg, and P. Vossen (eds.), *Euralex 1994 Proceedings, Papers submitted to the 6th EURALEX International Congress in Lexicography in Amsterdam,* The Netherlands. Amsterdam: Vrije Universiteit, 606-11.
Vossen, P. 2004. 'Eurowordnet: A Multilingual Database of Autonomous and Language-specific Wordnets Connected via an Inter-lingual-index.' *International Journal of Lexicography* 17(2): 161-187.