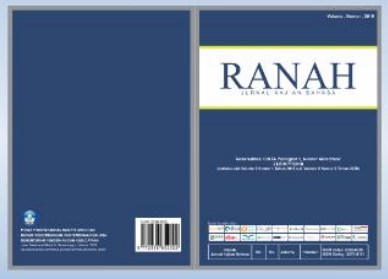

Ranah: Jurnal Kajian Bahasa
ojs.badanbahasa.kemdikbud.go.id/jurnal/index.php/jurnal_ranah

p-ISSN: 2338-8528
e-ISSN: 2579-8111

CURRENT IMPLEMENTATION AND FUTURE PROSPECTS OF SANTI-MORF V.1.0

Implementasi Terkini dan Prospek Masa Depan SANTI-morf v.1.0

Prihantoro*
Universitas Diponegoro
Jalan Prof. Sudarto, Tembalang, Kota Semarang, Jawa Tengah, Indonesia
prihantoro@live.undip.ac.id

Naskah Diterima Tanggal 1 Juni 2021—Direvisi Akhir Tanggal 12 November 2021—Disetujui Tanggal 19 Desember 2021
doi: https://doi.org/10.26499/rnh/v10i2.4189

Abstract

SANTI-Morf (Prihantoro, 2021) adalah sebuah program analisis morfologi terbaru untuk bahasa Indonesia. Dalam skema anotasi SANTI-morf (Prihantoro, A new tagset for morphological analysis of Indonesian, 2019), setiap token morfem terhubung dengan anotasinya. Token-token ini direpresentasikan dalam bentuk ortografis dan bentuk sitasi sehingga memungkinkan pengguna untuk melakukan penelusuran berbasis (alo)morf atau morfem. Selain itu, pengguna juga bisa melakukan penelusuran berbasiskan bentuk atau fungsi morfem. Ini karena tagset analitik yang digunakan di SANTI-morf mencakup bentuk (di antaranya: akar, klitik, jenis afiksasi) dan fungsi (di antaranya: aktif, pasif, derajat ajektiva). Saat ini, SANTI-morf diimplementasikan menggunakan NooJ (Silberztein, 2003), sebuah program pengembangan aplikasi linguistik. Pengguna dapat mengindeks dan menganotasi teks berbahasa Indonesia di komputer mereka, dan selanjutnya melakukan penelusuran menggunakan kriteria morfologi dan skema tokenisasi yang digunakan di skema anotasi SANTI-morf.

Kata-kata kunci: anotasi, penelusuran, morfologi, skema, SANTI-Morf, Nooj

Abstract

*SANTI-M*orf (Prihantoro, 2021) *is a new morphological analyser for Indonesian. In SANTI-Morf annotation scheme* (Prihantoro, 2019), *morpheme tokens are linked to their annotations. The tokens are presented in their orthographic and citation forms to allow (allo)morph or morpheme-based searches. Users can also perform retrievals on the basis of formal and functional morphological criteria as SANTI-Morf tagset encodes the analyses of morphemes' forms (e.g. roots, clitics, affix type) and functions (e.g. passive voice, active voice, adjective degrees, etc.). Currently, the scheme is implemented in Nooj* (Silberztein, 2003), *a linguistic development environment. It enables users to index and annotate Indonesian texts in their local PC, and later perform searches based on morphological criteria and or tokens defined by the SANTI-Morf scheme.*

Keywords: *annotation, retrieval, morphology, scheme, SANTI-Morf, Nooj*

How to Cite: Prihantoro. (2021). Current Implementation and Future Prospects of Santi-Morf V.1.0. *Ranah: Jurnal Kajian Bahasa*. *10*(2). 411—423. doi: https://doi.org/10.26499/rnh/v10i2.4189

*https://orcid.org/0000-0001-7708-9785

INTRODUCTION

There are a number of popular web services that index corpora in multiple languages including Indonesian corpora such as CQPweb Lancaster (Hardie, 2012)[1] or Sketch Engine[2] (Kilgarriff, et al., 2014). However, none of the Indonesian corpora in these web services is morphologically annotated. Two language-specific web services that offer Indonesian corpora which are morphologically annotated are Malay Concordance Project[3] (Gallop, 2013) and MalindoConc[4] (Nomoto, Akasegawa, & Shiohara, 2018). These two web services allow their users to access the morphological annotation presents in the corpora for testing hypotheses, validating claims, or supplying quantitative analyses, among many others research activities.

But what if the above-mentioned web services do not provide the annotated corpora required by the users? What if we want to morphologically analyse another corpus, such as a corpus in our local PC? Sketch Engine and CQPweb allow us to upload a corpus from a user's local PC, but no automatic annotation functionality is provided for Indonesian. A solution for this is to manually annotate our corpus, which is a reasonable approach when the size is small, around 10,000 words. But when the size of our corpus is relatively large, such as 500,000 words or more, it is more reasonable to automatically annotate the corpus using an automatic Morphological Annotation (MA) system.

This paper deals with the practical aspects of SANTI-morf, a new automatic MA system for Indonesian, namely how to install, activate, index text(s) from local PC as a corpus, and perform searches, using the morphological annotation scheme used in SANTI-morf. Other aspects (theoretical and computational) are discussed at length in Prihantoro (2021). SANTI is an acronym of *Sistem ANalisis Teks Indonesia* or in English, an annotation system for Indonesian texts. The *-morf* part is clipped from *morfem* 'morpheme'. The system allows users to index corpora kept in their local computers, tokenise each word in the corpora into morpheme tokens, and assign one or more morphological tags to each token.

SANTI-morf is presented here as an advancement of the existing MA systems for Indonesian, Two-Level Morphological Analyzer, thus TLMA[5] (Pisceldo, Mahendra, Manurung, & Arka, 2008) and MorphInd[6] (Larasati, Kuboň, & Zeman, 2011), both in terms of the system's implementation and the annotation scheme. SANTI-morf is implemented using NooJ (Silberztein, 2003), a platform that has been used to annotate various languages such as French, Turkish, Chinese, and Spanish among many others.

Users can access SANTI-morf using a graphical user interface, similar to many corpus analysis programs such as LancsBox[7] (Brezina, Timperley, & McEnery, 2018), AntConc[8] (Anthony, 2006), or WordSmith[9] (Scott, 1996). This differs from TLMA (Pisceldo et al., 2008) and MorphInd (Larasati et al., 2011) which are accessed via shell (terminal or command line). While accessing a program via shell is a method commonly used by programmers, linguists with minimum technical knowledge of programming may find this method challenging. For them, SANTI-morf offers a viable alternative.

[1] https://cqpweb.lancs.ac.uk/ (retrieved 18/11/2021)
[2] https://www.sketchengine.eu/ (retrieved 18/11/2021)
[3] https://mcp.anu.edu.au/ (retrieved 18/11/2021)
[4] https://malindoconc.lagoinst.info/concordance/ind/ (retrieved 18/11/2021)
[5] http://bahasa.cs.ui.ac.id/tools/MorphologicalAnalyzerIndonesia.zip (retrieved 18/11/2021)
[6] https://septinalarasati.com/morphind/ (retrieved 18/11/2021)
[7] http://corpora.lancs.ac.uk/lancsbox/ (retrieved 18/11/2021)
[8] https://www.laurenceanthony.net/software/antconc/ (retrieved 18/11/2021)
[9] https://www.lexically.net/wordsmith/ (retrieved 18/11/2021)

THEORETICAL BASIS

Manual and Automatic Annotation

Why should we bother annotating a corpus? An annotated corpus offers a number of benefits and eases for data analysis, information extraction, reusability, and reproducibility (McEnery, Xiao, & Tono, 2006, pp. 23-25) among many others. The annotation can be carried out manually or automatically.

Manual annotation is usually applied when the corpus is reasonably small, and when no automatic annotation system is available for the language or when the system cannot supply the analytic features required by the users. In some cases, the reasons could be manifold. For instance, Malihah (2013) preferred to manually annotate her corpus as (1) it is a reasonably small corpus, (2) no annotation system is available for Javanese, and (3) no automatic annotation system can encode functional grammatical features she studied. Some of the studies involving manual annotations are Gerstenberger et al. (2017) and Hu and Tan (2017), among many others.

This stands in contrast to other studies that require the analysis of a big corpus. Denistia & Baayen (2019), for instance, studied Indonesian allomorphs distributed over Leipzig Corpora Collection (LCC) Indonesian data[10], whose total size reaches millions of word tokens. Due to the big size of the corpus, in this case, it is more effective to carry out the annotation automatically. Love et al. (2017) and Prentice et al. (2011) are examples of studies, among many others, that also exploited annotated corpora. Note that it is also very common to combine both methods, for instance, by carrying out post-editing, or manual annotations, after the corpus is automatically annotated. As noted in the preceding section, SANTI-morf is an automatic annotation system. Thus, the annotation is carried out automatically.

Annotation Scheme

An annotation scheme, usually reflected by its tagset (a collection of analytic labels/tags), is neutral of system implementation. Let us illustrate this by comparing Penn Treebank[11] (Marcus, Marcinkiewicz, & Santorini, 1993) and CLAWS[12] (Garside, 1987) tagsets. While both are commonly used English tagset for POS (Part of Speech) tagging, the CLAWS tagset is more fine-grained overall.

For instance, mass and singular nouns receive only one tag in the Penn Treebank tagset. While the characteristics of these two features differ, a system that adheres to the Penn Treebank tagset will not be able to distinguish them, as it is not designed to do so. However, in the CLAWS tagset, these two analyses are expressed by two separate tags, hence two separate analyses. Therefore, a user who needs these two morphosyntactic features to be distinguished might prefer to use a system that adheres to the CLAWS rather than the Penn Treebank tagset.

While the abovementioned schemes are used for English, it is not fully compatible with Indonesian. In the next section, I concisely discuss SANTI-morf's morphological annotation scheme, as well as its implementation, as compared to other MA systems. SANTI-morf tagset, as a reflection of the annotation scheme, is presented in the DISCUSSION section.

SANTI-morf VS other MA Systems for Indonesian

To date, there are three automatic MA systems available for Indonesian: SANTI-morf, MorphInd, and TLMA. I here present SANTI-morf as an advancement of the other two systems built earlier, not only in terms of the implementation but also in terms of the annotation schemes to which they adhere.

[10] https://corpora.uni-leipzig.de/en?corpusId=ind_mixed_2013 (retrieved 18/11/2021)
[11] https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html (retrieved 18/11/2021)
[12] http://ucrel.lancs.ac.uk/claws7tags.html (retrieved 18/11/2021)

First, in terms of the annotation scheme, SANTI-morf decomposes words into full morpheme tokens. Thus, if a polymorphemic Indonesian word is composed of four morphemes, all the four morphemes are tokenised and presented in the output. This differs from TLMA in which only the word's root morpheme is supplied in the output. Second, SANTI-morf presents both the morphemes orthographic ((allo)morph) and citation (morpheme) forms when they differ. As for MorphInd, only the citation form is presented in the output. Third, SANTI-morf can handle polymorphemic words produced using affixation, reduplication, compounding, and cliticisation. In this case, MorphInd is equally powerful to SANTI-morf as it can also analyse those four morphological processes. This stands in contrast to TLMA, which cannot analyse compounding and cliticisation. Fourth, SANTI-morf corresponds all tokens to morphological tags. In MorphInd, conversely, affixes are left unannotated. Fifth, SANTI-morf includes *formal* morphological analytic categories, up to sub-categories. For instance, affixes are further sub-categorised into prefix, suffix, infix, and circumfix. Reduplication is further sub-categorised into full, partial, and imitative reduplication. TLMA supplies only reduplication category without further sub-categorising it, while MorphInd totally excludes formal morphological analyses in its scheme. As for the *functional* analytic categories, SANTI-morf encodes fine-grained analyses. For instance, unlike TLMA which only has three POS categories (noun, verb, adjective), SANTI-morf encodes 12 POS categories (for root – presented in the subsequent section). SANTI-morf also includes functional categories such as reflexive and reciprocal, which are absent in MorphInd. In addition, SANTI-morf functional categories are fully driven by Indonesian reference grammars, namely Alwi et al. (1998) and Sneddon et al. (2010). When an annotation scheme is driven by reference grammars, or other equivalent resources, in the same target language, it can better reflect the analytic categories actually used in the language targeted by the system, in this case, Indonesian. Thus, users needs to perform searches based on these categories can reasonably be anticipated. Conversely, MorphInd tagset is to some extent inspired by Penn Treebank (Larasati, Kuboň, & Zeman, 2011, p. 122), an English tagset, as discussed earlier. Thus, some features such as singular and plural, even for verbs, are encoded. These analytic categories should have been unspecified as no number agreement exists in Indonesian (Prihantoro, 2021, p. 292).

In terms of implementation, SANTI-morf has a mechanism to deal with the out-of-vocabulary (i.e. unknown words) problem. Therefore, it allows the system to deal with words the system cannot recognise due to the paucity of resources. Among many others, proper names, orthographic variations, misspelt words, newly coined words are likely to be examples of such words. In TLMA and MorphInd, these words will be left unanalysed (or tagged as unknown). The coping mechanism in SANTI-morf allows the system to consistently produce 100% coverage. In terms of the evaluation, SANTI-morf can reach 99% precision and recall with only 1% ambiguity rate when tested on a testbed corpus. MorphInd does not produce any ambiguous output, but its accuracy is measured less than 90% (due to a large number of unknown words in the testbed corpus).

RESEARCH METHOD

The creation of SANTI-morf can be summarised into four steps. The first step is the creation of the morphological annotation scheme, whose output is a morphological annotation tagset for use in SANTI-morf. Each analytic tag is a combination of formal (the type of morpheme) and functional (the function of the morpheme) analytic labels. A tag must have a main formal category label; it can be followed by its subcategory (marked +), or one or more functional category labels (also marked +). An underscore (_) is incorporated into each outcome POS analytic label. This analytic category marks POS of a word, a morpheme can mark. Thus,

it is a resulting POS (thus starts in R) marked by a morpheme. The R_ precedes each of these analytic labels to distinguish outcome POS from root POS (not marked by R_).

Table 1.
Formal and functional analytic labels used in SANTI-morf tagset

Formal	Functional
ROOT: root	+ACV: active
+PCLT: proclitic	+PSV: passive
+ECLT: enclitic	+RECP: reciprocal
+LOST: root with first consonant deletion	+RFLX: reflexive
PFX: prefix	+APPL: applicative
SFX: suffix	+CAUS: causative
IFX: infix	+EQTV: equative degree
CFX: circumfix	+ITRV: iterative aspect
+A: opening circumfix element	+RAND: random unordered event
+Z: closing circumfix element	+DEF: definite = *nya*
RED: all reduplication	+NYA: depend on how = *nya* function
+FULL: full reduplication	+SPV: superlative degree
+PART: partial reduplication	+ADJ: adjective root morpheme
+IMTV: imitative reduplication	+ADV: adverb root morpheme
	+ART: article root morpheme
	+CLS: classifier root morpheme
	+CNJ: conjunction root morpheme
	+ITJ: interjection root morpheme
	+NOU: noun root morpheme
	+NUM: numeral root morpheme
	+PCL: particle root morpheme
	+PRE: preposition root morpheme
	+PRO: pronoun root morpheme
	+VER: verb root morpheme
	+FRG: foreign root morpheme
	+R_ADJ: adjective outcome POS morpheme
	+R_ADV: adverb outcome POS a morpheme
	+R_NOU: noun outcome POS morpheme
	+R_VERB: verb outcome POS morpheme
	+R_NUM: numeral outcome POS morpheme

The second step is the selection of a platform to implement SANTI-morf. There are at least three platforms that can potentially be used to apply SANTI-morf, namely xfst[13] (Beesley & Karttunen, 2003), foma[14] (Hulden, 2009), and NooJ (Silberztein, 2003). As Larasati et al., (2011, p. 24) mentioned, the *compile-replace* function in xfst is patent-encumbered; thus, they used foma, a free platform to develop a morphological annotation system. However, foma does not have any built-in disambiguation function (a possible workaround is available, but quite complex to implement by non-programmers). Disambiguation is required as some morphemes are contextually ambiguous such as *-an* which may be a nominaliser suffix, or a part of a nominaliser circumfix as in *ke—an*, or *per—an.* I prefer to implement SANTI-morf in NooJ, as it is completely free, supports disambiguation, and it also provides a corpus query function, similar to corpus analysis programs typically used by linguists.

The third step is the creation of SANTI-morf morphological annotation resources. The created resources are lexicons and rules (morphotactic, morphophonemic, and disambiguation), discussed at length in Prihantoro (2021). Fourth, the performance of the system is evaluated, whose output is a configuration file that organises the resources in a way that provides the best possible output. For the testbed, I created a 10,000-word corpus whose data is randomly curated from Leipzig Corpora Collection (Goldhahn, Eckart, & Quasthoff, 2012). Precision and recall (Ting & Geoffrey, 2011, p. 781) are used in the evaluation as SANTI-morf can produce ambiguous results. In this context, precision can briefly be described as the proportion of correct

[13] https://web.stanford.edu/~laurik/.book2software/ (retrieved 18/11/2021)
[14] https://fomafst.github.io/ (retrieved 18/11/2021)

annotations relative to all existing annotations. As for recall, it is the proportion of correct annotations relative to the sum of units correctly annotated and units left unannotated.

Upon several experimentations, the best performance is achieved by organising the resources in four modules running in pipeline: the Annotator (carry out initial annotations), the Guesser (analyse unknown words), the Improver (add correct annotations to units deemed incorrectly annotated by the previous two modules), and the Disambiguator (resolve ambiguities). The best performance here refers to the highest precision and recall, as well as the lowest ambiguity, achieved by the system. SANTI-morf scores 99% for precision and recall with 1% ambiguity rate, as noted earlier in the preceding section.

DISCUSSION

As noted earlier in the INTRODUCTION section, this paper seeks to describe the practical aspects of SANTI-morf, i.e, its implementation for end-users. To fulfil this aim, the architecture of the system is not discussed here. Instead, I focus on explaining how to install and activate the system, index text(s) in our local PC as a corpus, and how to perform a variety of searches using morphological criteria defined in SANTI-morf's annotation scheme discussed in the preceding section.

Installation

SANTI-morf is implemented using NooJ. Thus, the first step is to install NooJ on our local computer. The installation file can be obtained from the NooJ download page[15]. NooJ video tutorials[16] are available in several languages. The tutorials in Indonesian also include a how-to-install video. Once the installation is completed, NooJ's graphical user interface will appear on your screen.

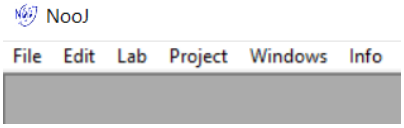

Figure 1.
NooJ's graphical user interface

SANTI-morf activation

By default, NooJ can only support English. An extra step is required to enable supports for other languages, including Indonesian. To do this, go to Info, and choose Preferences. Next to Language Name, choose id (ISO 639-1 code for Indonesian), and click download module. This means we ask NooJ to automatically download all resources for the Indonesian language from the NooJ official repository to our local computer.

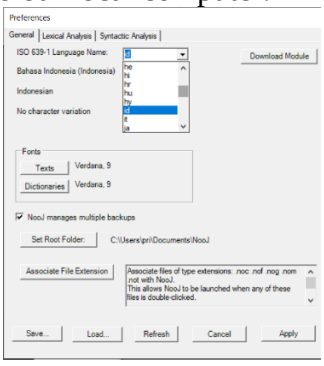

Figure 2
SANTI-morf Activation via NooJ's Preferences

[15] http://www.nooj-association.org/downloads.html (retrieved 18/11/2021)
[16] http://www.nooj-association.org/tutorials.html (retrieved 18/11/2021)

Once the download is completed, still in Preferences, click the Load button at the lower area of the Preferences. A dialogue box will appear. Go inside the id directory and choose SANTI-morf_v20201209.noj; this is SANTI-morf's configuration file. Then, click open. Next, in Preferences, click Apply (down right corner). This activates all SANTI-morf functionalities.

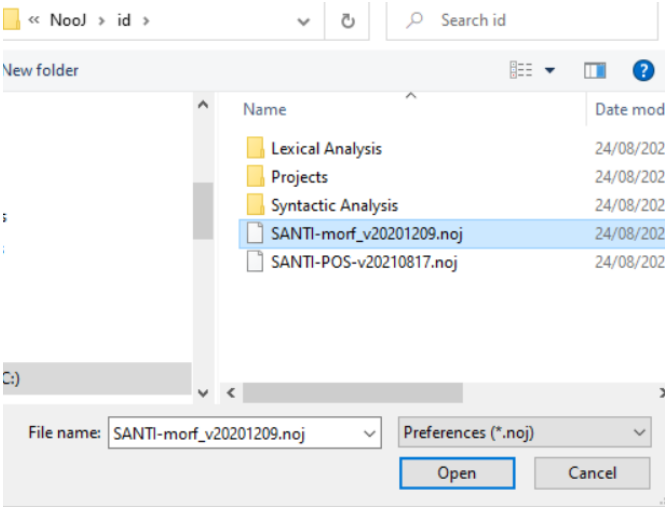

Figure 3.
SANTI-morf's configuration file

Indexing

Indexing here means to load the corpus onto the NooJ platform. There are various ways to index a corpus in NooJ. In some cases, a corpus can be composed of a single text file, kept somewhere on our PC. To index this kind of corpus, click File, Open, and choose Text. Then, find the directory where the corpus is kept, and choose the file.

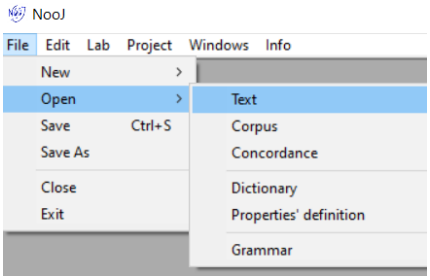

Figure 4.
Using a text file as a corpus

A corpus creation window will pop up, which allows us to: choose the language of the corpus, specify the corpus file format that we have, and set Text Unit. If our corpus is raw and in the format of .txt, in most cases, we can directly just click OK (down right corner). Modifications to the setting are relevant when our corpus is in non .txt format (.docx, .pdf, or .html among many others) or when our corpus is already annotated.

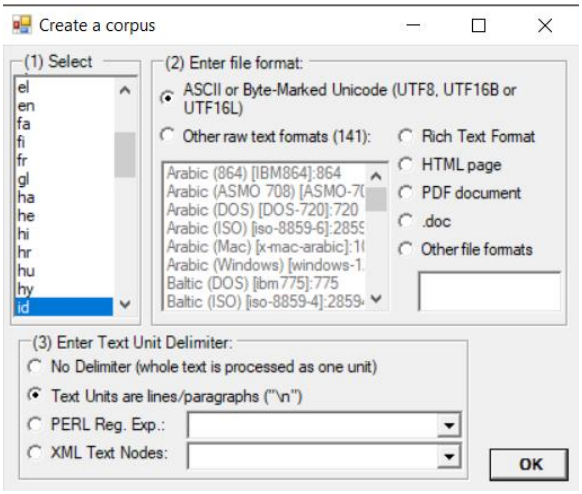

Figure 5.
Corpus Creation Window

In some other cases, a corpus can be composed of multiple text files kept on a local PC. To index multiple files as a corpus, click File, New, and choose Corpus. Name the corpus and choose Add to select multiple text files we wish to add as the corpus texts.

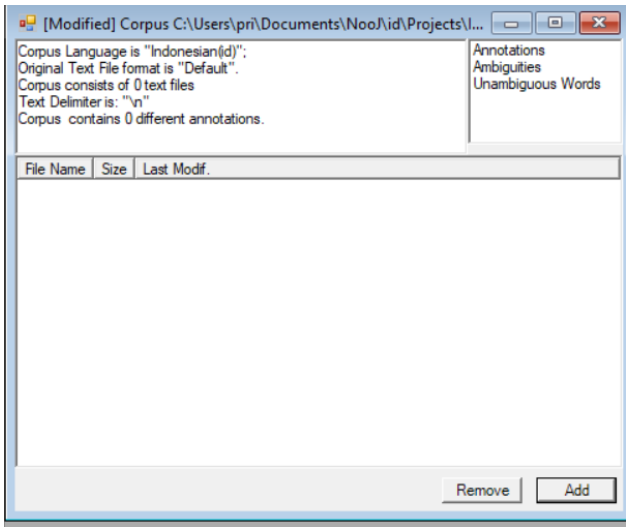

Figure 6.
Using multiple files as a corpus

SANTI-morf annotation

Once the corpus is indexed, it is ready for the annotation process. Right-click anywhere on the corpus. A panel will pop up. Subsequently, click Linguistic Analyses and wait until the annotation is completed.

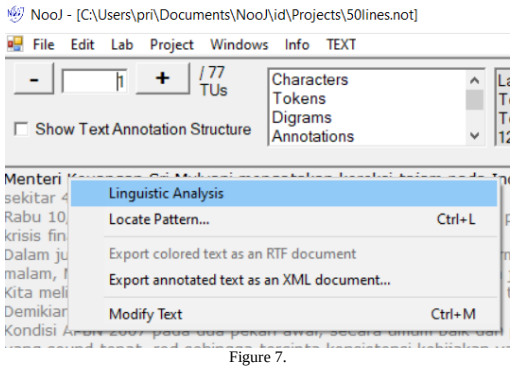

Figure 7.
Corpus annotation

Corpus Query

Once the annotation is completed, we can now build and send queries. I here demonstrate some of the queries. To display the query window, press CTRL+L. Alternatively, right-click on anywhere on the corpus, and choose Locate Pattern. A query window will pop up. Queries can be written in the query box (under a Nooj Regular Expression).

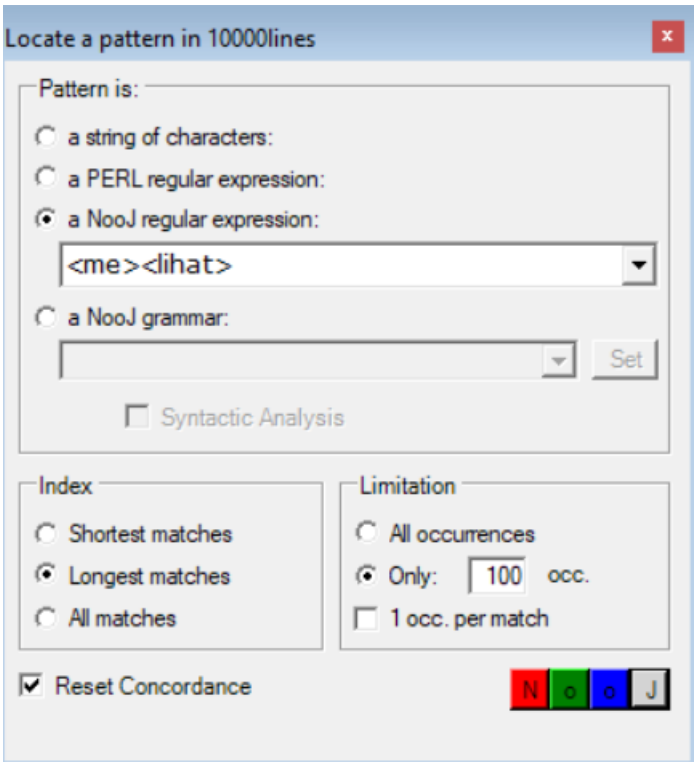

Figure 8.
Corpus Query Window

In the query, each morpheme token must be surrounded by angle brackets, e.g. <me><lihat>. Once we are happy with our query, click one of the colourful buttons at the down right corner of the query window. This retrieves all words which contain a combination of active verb prefix *me-* and verbal root morpheme *lihat* 'to look'. `

Table 2.
Randomly Selected Concordance Lines from the Query: <me><lihat>

Before	Sequence	After
justru membaik. Kita	*melihat*	*faktor inflasi dari*
misalnya. Kita tak	*melihat*	*dampak yang terlalu*
pertemuan itu, Dubes	*melihat*	*banyak hal yang*
ujar Dubes yang	*melihat*	*banyak hal yang*
bisa berubah dengan	*melihat*	*angka perkembangan dari*

All morpheme queries are written naturally in terms of their order, except for infix, whose query is written like a prefix. Thus, if a root is specified, the query would be <jari>, in which the infix precedes the root. This would give *jemari* 'fingers' in the result.

When using tags, the main formal analytic label (ones that do not begin with + in the tagset) must be used and can be followed by its subcategories or functional category labels. Therefore, <PFX> or <PFX+R_VER> is a valid query because they all begin with PFX (prefix),

one of the main formal analytic labels, but not <R_VER>, <+R_VER>, or <R_VER+PFX>, because +R_VER(verb outcome) is a functional category. The query <PFX+R_VER> retrieves words containing prefix morphemes whose outcomes are all verbs, regardless of the form (*ber-*, *di-*, *mem-*, among many others).

Table 3.
Randomly Selected Concordance Lines from the Query: <PFX+R_VER>

Before	Node	After
Pemerintah juga telah	*berkomitmen*	*untuk meningkatkan produksi*
persen menjadi 88.343 ton	*dibanding*	*September yang hanya*
Ekonomi Bangsa, yang	*digelar*	*8 Juli mendatang di*
ekonomi Indonesia justru	*membaik*	*. Kita melihat faktor*
ada dalam upaya	*membantu*	*pengusaha Indonesia yang*

If forms are not specified, the formal category label slot can be replaced by ALU (Atomic Linguistic Unit), a NooJ wild card label for any token or category. Thus, inserting <ALU+R_NOU> will give us all words containing all morphemes, regardless of the formal category, whose outcome is a noun. As the formal category is unspecified, the formal category of such morphemes may vary (nominaliser prefix *peng-*, nominaliser circumfix *ke—an* nominaliser suffix *-an*, etc.).

Table 4.
Randomly Selected Concordance Lines from the Query: <ALU+R_NOU>

Before	Node	After
tersebut sudah mempunyai	*keinginan*	*menambah lima pesawat*
kita lihat dalam	*kejadian*	*'subprime mortgage' misalnya*
Dubes saat menerima	*pengurus*	*ICMI London yang*
menjadi jembatan antara	*pengusaha*	*Indonesia dengan mitranya*
M Natalegawa mengharapkan	*Ikatan*	*Cendekiawan Muslim Indonesia*

We can combine orthographic form and tags in the query. For instance, we can insert <per,PFX>. This will retrieve all words with *per-* as their prefixes, not as a part of an opening element of a circumfix.

Table 5.
Randomly Selected Concordance Lines from the Query: <ALU+R_NOU>

Before	Node	After
tetap 950 ribu barel	*perhari*	*. Oleh karena itu*
semua akan kita	*perkuat*	*. Itu memakan waktu*
besar itu, satu	*persatu*	*, jelasnya. Benchmark tersebut*
I 2007 semata-mata	*diperoleh*	*dari kegiatan operasional*
rencana bisnis untuk	*mempercepat*	*perkembangan Bank Mandiri*

It is also possible to retrieve morphemes via their citation forms (when different from the orthographic form). To do that, insert the formal category label of the morpheme (or ALU if unspecified), followed by the citation form. For instance, the query <PFX+meN> or <ALU+meN> will retrieve all words containing all allomorphs of *meN-* in the corpus.

Table 6.
Randomly Selected Concordance Lines from the Query: <ALU+meN> or <PFX+meN>

Before	Node	After
tersebut, Medco akan	*memasok*	*gas dalam kurun*
ekonomi Indonesia justru	*membaik*	*. Kita melihat faktor*
tahun ini berencana	*menambah*	*lima pesawat terbang*
ICMI London dalam	*mengisi*	*peluang yang ada*
negatif karena mampu	*menyerap*	*kerugiannya dengan menggunakan*

In some cases, the root's first consonant is deleted due to morphophonemic processes as in *men(t)ingkat* 'to improve (intr)'. To identify roots whose first consonant is deleted, the query <ROOT+Lost> can be used. Alternatively, use <ALU+Lost> when the formal category of the morpheme is not specified.

Table 7.
Randomly Selected Concordance Lines from the Query: <ALU+Lost> or <PFX+Lost>

Before	Node	After
keyakinan kepada investor.	*Menurut*	*data Depkeu, net*
efektif Januari, ujarnya.	*Pemerintah*	*juga telah berkomitmen*
itu disampaikan dubes saat	*menerima*	*Pengurus ICMI London*
ICMI London dalam	*mengisi*	*peluang yang ada*
tersebut, Medco akan	*memasok*	*gas dalam kurun*

The results are also equipped with various statistical elements, which can be used to incorporate quantitative analyses when interpreting the results. This can help with hypothesis testing. For instance, one may hypothesise that *meng-* is the most productive allomorph as compared to the other allomorphs of *meN-*. SANTI-morf can help test this hypothesis. Some are demonstrated here. I ran SANTI-morf on BBPT-PAN Indonesian corpus (Adriani & Hamam, 2009); the frequencies of *meng-*, *meny-* and *men-*, *mem-*, *me-*, *menge-*, all allomorphs of *meN-*, are 8947, 2104, 12577, 6835, 7757, and 47, respectively. We here see that the hypothesis is rejected as the most frequent allomorph is *men-*. Users can extend the analysis to other corpora; they can also carry out different experimentations using other statistical software supports.

CLOSING

This paper has fulfilled the aim presented earlier in the introduction section, that is, to introduce the practical aspects of SANTI-morf. The steps on how to install NooJ, index a corpus, activate SANTI-morf, annotate a corpus, and writing various queries based on forms and or morphological users wish to study have been demonstrated. This indicated that SANTI-morf, as I claimed in the introduction section, may help with hypothesis testing, validation of linguists' introspection, and finding answers to research questions, particularly for linguists who wish to carry out a corpus-based morphological study. While not all features of SANTI-morf are demonstrated in this paper, due to the words limit, I argue that readers will find this paper informative and useful.

BIBLIOGRAPHY

Adriani, M., & Hamam, R. (2009). *Research Report Phase 3.2: Final Report on Statistical Machine Translation for Bahasa Indonesia - English and English to Bahasa Indonesia.* Jakarta: BPPT.

Alwi, H., Dardjowidjojo, S., Lapoliwa, H., & Moeliono, M. (1998). *Tata Bahasa Baku Bahasa Indonesia (3rd Edition).* Jakarta: Balai Pustaka.

Anthony, L. (2006). Concordancing with AntConc: An introduction to tools and techniques in corpus linguistics. *JACET Newsletter*, 155-185.

Beesley, K. R., & Karttunen, L. (2003). *Finite State Morphology*. Stanford: CSLI.

Brezina, V., Timperley, M., & McEnery, T. (2018). *#LancsBox v. 4.x [software]*. Available at: http://corpora.lancs.ac.uk/lancsbox

Denistia, K., & Bayeen, H. (2019). The Indonesian prefixes PE-and PEN-: A study in Productivity and Allomorphy. *Morphology 29(3)*, 385-407. https://doi.org/10.1007/s11525-019-09340-7

Gallop, A. T. (2013). The language of Malay manuscript art: a tribute to Ian Proudfoot and the Malay Concordance Project. *International Journal of the Malay World and Civilisation 1(3)*, 11-27.

Garside, R. (1987). The CLAWS Word-tagging System. In R. Garside, G. Leech, & G. Sampson (eds.), *The Computational Analysis of English: A Corpus-based Approach* (pp. 31-41). London: Longman.

Gerstenberger, C., Partanen, N., & Rießler, M. (2017). Instant annotations in ELAN corpora of spoken and written Komi, an endangered language of the Barents Sea region. *Proceedings of the 2nd Workshop on the Use of Computational Methods in the Study of Endangered Languages*, (pp. 57-66). https://doi.org/10.18653/v1/W17-0109

Goldhahn, D., Eckart, T., & Quasthoff, U. (2012). Building Large Monolingual Dictionaries at the Leipzig Corpora Collection: From 100 to 200 Languages. In N. Calzolari, K. Choukri, T. Declerck, M. Doğan, B. Maegaard, J. Mariani, & S. Piperidis (Eds.), *Proceedings of LREC Vol. 29* (pp. 31-43). Istambul: European Language Resources Association (ELRA).

Hardie, A. (2012). CQPweb—combining power, flexibility and usability in a corpus analysis tool. *International journal of corpus linguistics, 17(3)*, 380-409. https://doi.org/10.1075/ijcl.17.3.04har

Hu, C., & Tan, J. (2017). Using UAM Corpus tool to Explore the Language of Evaluation in Interview Program. *English Language Teaching, 10(7)*, 8-20. https://doi.org/10.5539/elt.v10n7p8

Hulden, M. (2009). Foma: a Finite-State Compiler and Library. In A. Lascarides (Ed.), *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL '09)* (pp. 29-32). Stroudsburg, PA: EACL. https://doi.org/10.3115/1609049.1609057

Kilgarriff, A., Baisa, V., Bušta, J., Jakubíček, M., Kovář, V., Michelfeit, J., & Suchomel, V. (2014). The Sketch Engine: ten years on. *Lexicography (1) 1*, 7-36. https://doi.org/10.1007/s40607-014-0009-9

Larasati, S.-D., Kuboň, V., & Zeman, D. (2011). Indonesian Morphology Tool (MorphInd): Towards an Indonesian Corpus. *Systems and Frameworks for Computational Morphology* (pp. 119-129). Zurich: Springer. https://doi.org/10.1007/978-3-642-23138-4_8

Love, R., Dembry, C., Hardie, A., Brezina, V., & McEnery, T. (2017). The Spoken BNC2014: Designing and building a spoken corpus of everyday conversations. *International Journal of Corpus Linguistics, 22(3)*, 319-344. https://doi.org/10.1075/ijcl.22.3.02lov

Marcus, M. P., Marcinkiewicz, M. A., & Santorini, B. (1993). Building a Large Annotated Corpus of English: The Penn Treebank. *Computational linguistics 19(2)*, 313–330. https://doi.org/10.21236/ADA273556

McEnery, T., Xiao, R., & Tono, Y. (2006). *Corpus-based language studies: An advanced resource book*. Milton Park: Taylor & Francis.

Nomoto, H., Akasegawa, S., & Shiohara, A. (2018). Building an open online concordancer for Malay/Indonesian. *Paper presented at the 22nd International Symposium on Malay/Indonesian Linguistics (ISMIL)*. University of California, Los Angeles.

Pisceldo, F., Mahendra, R., Manurung, R., & Arka, I. W. (2008). A Two Level Morphological Analyser for the Indonesian Language. In N. Stokes, & D. Powers (Eds.), *Proceedings of Australasia Technology Association Workshop* (pp. 142-150). Hobart: ACL.

Prentice, S., Taylor, P. J., Rayson, P., Hoskins, A., & O'Loughlin, B. (2011). Analyzing the semantic content and persuasive composition of extremist media: A case study of texts produced during the Gaza conflict. . *Information Systems Frontiers, 13(1)*, 61-73. https://doi.org/10.1007/s10796-010-9272-y

Prihantoro. (2019). A new tagset for morphological analysis of Indonesian. *International corpus linguistics conference*. Cardiff.

Prihantoro. (2021). *A new morphological annotation system for Indonesian (PhD Thesis).* Lancaster: Lancaster University Press.

Prihantoro. (2021). An Evaluation of the Morphological Annotation Scheme for Indonesian Used in MorphInd Program. Corpora, in Press. *Corpora 16 (3)*, in press. https://doi.org/-10.3366/cor.2021.0221

Scott, M. (1996). *WordSmith manual*. Gloucestershire: Lexical Analysis Software ltd.

Silberztein, M. (2003). *NooJ Manual.* Available for download at: www.nooj4nlp.net.

Sneddon, J. N., Adelaar, A., Djenar, D.-N., & Ewing, M.-C. (2010). *Indonesian Reference Grammar:2nd Edition.* New South Wales: Allen & Unwin.

Ting, K. M., & Geoffrey, W. (2011). Precision and Recall. In S. C, *Encyclopedia of Machine Learning* (p. 781). Boston: Springer.