Usability improvements in the lexicographic framework Toposław

Marcin Woliński,[*] Agata Savary,[†*] Piotr Sikora,[*] Małgorzata Marciniak[*]

[*]Institute of Computer Science, Polish Academy of Sciences,
J.K. Ordona 21, 01-237 Warszawa, Poland
{wolinski, mm}@ipipan.waw.pl

[†] Université François Rabelais Tours,
3, pl. Jean-Jaurès, 41029 Blois, France
agata.savary@univ-tours.fr

Abstract

The paper concerns a dictionary of urban proper names where graphs are used for the representation of inflectional and pragmatic variants of names. We describe enhancements of the *Multiflex* formalism concerning alternative values of features, abbreviations and letter case. Then we present improvements of our tool *Toposław* in graph management and debugging.

Keywords: electronic dictionary, urban proper names, lexicographic framework, graphs, Polish

1. Introduction

We present ideas on how to make lexicographic work easier during the creation of an electronic dictionary of urban proper names. Now, the dictionary contains 7832 Warsaw names: streets, monuments, buildings, etc. One of its main goals is the recognition and generation of all grammatical forms and variants of a proper name, including transliterated spoken variants. For example, the full official name of *ulica Bitwy Warszawskiej 1920 r.* (The Battle of Warsaw 1920 Street) is abbreviated in practice into *ul. Bitwy Warszawskiej* (The Battle of Warsaw Str), while for transliterated speech processing the year should be spelled out: *tysiąc dziewięćset dwudziestego roku* (year nineteen twenty). Our dictionary describes 132217 forms.

To help build the dictionary we created our own tool *Toposław* which cooperates with *Morfeusz* (Woliński, 2006), a morphological analyser and generator for Polish words, and *Multiflex* (Savary, 2005), a cross-language, morpho-syntactic generator of multi-word units built upon *Unitex* (Paumier, 2003). In (Marciniak et al., 2009) we presented the assumptions made during the creation of the dictionary, tools used, and data prepared for input. The first version of the program described in (Sikora and Woliński, 2009) was the starting point of the lexicographer's work. On the basis of these first experiences we identified several obstacles to effective lexicographic work: the growing number of graphs describing proper names, their complexity, and the need for graph reusing facilities. Solutions to these problems are presented in the present paper.

Our dictionary is meant for knowledge-based analysis and generation of written and spoken Polish texts. (Piskorski et al., 2009) show that knowledge-poor methods give good results in matching two-component person names in Polish texts (up to 0.99 accuracy in some cases). An automatic lemmatization of such names is more problematic (below 0.67 accuracy in case of component inversion). We expect such methods to be even less effective for urban names, allowing for omissions, inversions and other transformations. We believe that in order to achieve high-quality analysis and generation of urban names we need to use knowledge-based methods.

(Abramowicz et al., 2006) describe a Polish named-entity extraction tool, based on rich gazetteers (164,000 names), local grammars (198 rules) and string-based similarity. It allows for the population of an ontology from free text. Our approach is complementary in the sense that we describe not only morphological but also syntactic and semantic variants, and we point out pragmatic variants necessary for text generation. Our grammars (i.e. graphs) are fully lexicalized, which avoids both noise and silence.

2. Enhanced Multiflex Formalism

Our dictionary should describe all variants of collected proper names. Example (1) illustrates variants of a person name consisting of a first and a last name, while (2) gives all variants of the same name with potential information on the military rank — general.

(1) Władysław Anders / Wł. Anders / W. Anders / Anders

(2) generał Władysław Anders / generał Wł. Anders / generał W. Anders / generał Anders / gen. Władysław Anders / gen. Wł. Anders / gen. W. Anders / gen. Anders / Władysław Anders / Wł. Anders / W. Anders / Anders

Battlefield heroes quite often have pseudonyms that can appear before or after the last name, and can be in quotes or not. Thus the variants of a proper name can be numerous and their description poses a significant challenge to the lexicographer.

2.1. Pragmatic Labels

Among many variants of a name, we want to distinguish a few important pragmatic variants marked by specific values of the *Usage* category:

- the official variant (*offic*) used in official lists and documents,
- the neutral variant (*neut*) preferred in text generation,

- the neutral spoken variant (*spok*) preferred for speech generation.

Figure 1 shows the morphological analysis and the inflection graph of the compound person name whose variants are shown in (1). The middle path describes the full form containg the first and the last name (*Władysław Anders*). The two names inflect and agree for case, which is reflected by the common unification variable *$c*. The resulting compound forms inherit their number and gender from the first constituent and have the conforming case ⟨*Gen=$1.Gen;Nb=$1.Nb;Case=$c*⟩. They are either considered as official variants of the name, or they remain unmarked for the usage variant, (⟨*Usage=offic*|⟨*E*⟩⟩ in the output under the path, where ⟨*E*⟩ denotes an empty value). The upper path describes the elliptical variant *Anders* annotated as the neutral (*neut*), the neutral spoken (*spok*) or unmarked variant (⟨*E*⟩). The bottom path allows us to obtain the initial of the first name followed by the family name, here *Wł. Anders*, unmarked for usage.

Note that the neutral and spoken variants are identical, which is the case for many names. In order to represent them with one path, we enhanced the formalism presented in (Marciniak et al., 2009) with the possibility of expressing alternative values in a feature structure. In Fig. 1 the alternative operator 'l' allows us to reduce the graph's size from 7 to 3 paths.

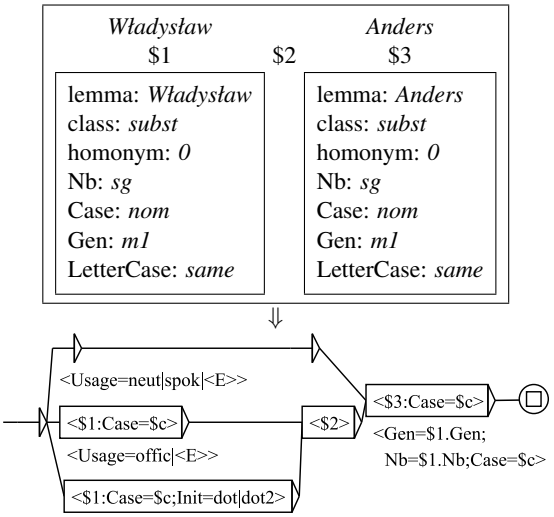

Fig. 1: Inflection graph for *Władysław Anders*

The proper name variants from (2) are described by the graph in Fig. 2. Its third component (label $3) is one of the variants from example (1), described by Fig. 1. Usage variants attached to paths of the embedded graph are available to the external graph due to the fact that they are described as output values.[1]

2.2. Initials

Urban proper names frequently take abbreviated forms of their components when appearing in written texts. Any first name can be reduced to its one or two initial letters

[1]Output values are written under boxes.

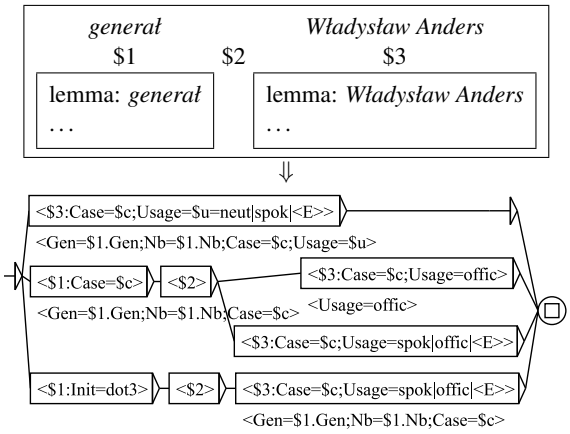

Fig. 2: Use of pragmatic labels for embedded components

followed by a dot as in example (1). Similarly behave the words 'Street', 'Square', 'Avenue', titles and functions, such 'general' in example (2). Using a dictionary of abbreviations would be most appropriate, unfortunately we are not aware of such a dictionary for Polish. Thus we propose the following partial solution. Whenever an abbreviation is constructed from one to five initial letters, whether or not followed by a dot, the category *Init* with the corresponding value can be used as shown for component *$1* in Fig. 1 (*Init=dot2* for *Władysław — Wł.*) and in Fig. 2 (*Init=dot3* for *generał — gen.*).

Note that some words are abbreviated differently than by a prefix, as in *płk* for *pułkownik* 'colonel'. Moreover abbreviations or acronyms may be formed from inflected words as *W-wie* for *Warszawie*, or may become independent inflecting lexemes, e.g. *ONZ*, *ONZ-u* for *Organizacja Narodów Zjednoczonych* 'United Nations'. Such cases are currently described with specific dedicated inflectional graphs.

2.3. Features

For making graphs in *Multiflex* clearer, we distinguish three categories of features (compare Fig. 3 with Fig. 1 in (Savary et al., 2009)):

- features shared with the underlying module for simple words, here *Morfeusz*, such as number, gender, case, etc. (see ⟨CATEGORIES⟩)
- features freely defined by the user on the level of multi-word units, such as pragmatic labels (see ⟨EXTRA_CATEGORIES⟩)
- features implemented in *Multiflex* in order to handle morpho-graphical problems, such as letter case, initialisms and acronyms (see ⟨GRAPHICAL_CATEGORIES⟩)

The first version of our tool suffered from letter case disagreement between simple and compound words. To cope with this issue we introduced the *LetterCase* category (see Fig. 4) mentioned in Fig. 3. It indicates how to transform the letter case of the lemma into the form desired in the dictionary. If no transformation is needed the

⟨**CATEGORIES**⟩

Nb:	sg , pl
Case:	nom, gen, dat, acc, inst, loc, voc
Gen:	m1, m2, m3, f, n1, n2, p1, p2, p3
Pers:	pri, sec, ter
Deg:	pos, com, sup
Asp:	imperf, perf
...	

⟨**EXTRA_CATEGORIES**⟩

Usage:	⟨E⟩,offic, neut, spok

⟨**GRAPHICAL_CATEGORIES**⟩

LetterCase:	all_lower, all_upper, first_upper, same first_upper_each_word
Init:	⟨E⟩,dot,no_dot,dot2,no_dot2,dot3,no_dot3 dot4, no_dot4, dot5, no_dot5

⟨CLASSES⟩

subst:	(Nb,⟨var⟩),(Case,⟨var⟩),(Gen,⟨fixed⟩)
adj:	(Nb,⟨var⟩),(Case,⟨var⟩),(Gen,⟨var⟩),(Deg,⟨var⟩)
...	

Fig. 3: Three kinds of categories in the morphological model of Polish in *Multiflex*

Form	**Lemma**	**LetterCase**
empik	*EMPiK*	*all_lower*
STUDIO	*studio*	*all_upper*
Długa	*długi*	*first_upper*
Centrum handlowe	*centrum handlowe*	*first_upper*
Centrum Handlowe	*centrum handlowe*	*first_upper_each_word*
Władysław	*Władysław*	*same*

Fig. 4: Values of the *LetterCase* category

value is *same*. The *LetterCase* value is most often implicit, i.e. it does not appear in inflection graphs but is automatically deduced from each component of a compound during its morphological analysis. Its utility is most visible when a common word is a component of a proper name. For instance in Fig. 5 the lemma of *Długa* is a common adjective spelled in lowercase as an individual word, and in uppercase as a part of a street name. Thus it obtains a morphological analysis in which the values stemming from *Morfeusz* are automatically completed by the *first_upper* feature.

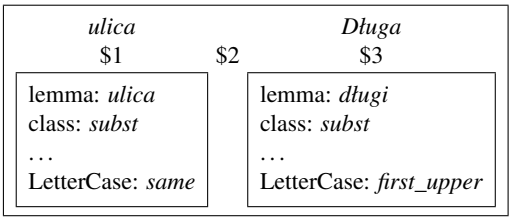

Fig. 5: Explicit letter case values in *ulica Długa*

3. Improvements in *Toposław*

Describing multi-word proper names with many variants requires a lot of work. We support this task with dedicated tools, similarly to other related projects, e.g. Ws4LR (Krstev et al., 2006).

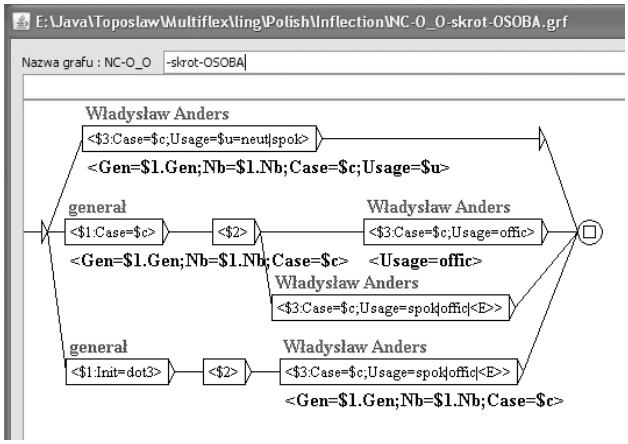

Fig. 6: Boxes in a graph labelled with components of the compound *generał Władysław Anders*

3.1. Graph Management

As the number of graphs grows with the number of names described, managing graphs becomes difficult. Currently we have over 7832 names with 405 corresponding graphs. The majority of names use only a few graphs, which are thus easy to remember. For the rest, however, the user needs some support.

When a lexicographer introduces a new proper name, *Toposław* displays the list of currently defined graphs which have the same number of components as the name in question. This significantly reduces the number of graphs that have to be considered.

Moreover in the process of describing a compound, the user is asked to state for each component whether it inflects or not. This pattern of inflecting components is again used to filter the list of graphs. Namely, a graph matches a name only if:

- for each component marked as inflecting there exists a corresponding box in the graph marked as inflecting (with an equality on some grammatical feature),
- for each component marked as non-inflecting none of the corresponding boxes allow for inflection.

We plan to extend this mechanism by measuring the morphological similarity of components of a name being considered to the components of names already described. This measure will allow us to rank the graphs and suggest the most promising graph for a given name (see (Krstev and Vitas, 2009) for other graph prediction facilities).

Another important aid, which allows the user to check whether a graph matches, is provided by labeling boxes in the graph with constituents of the compound (cf. Fig. 6). When the user selects a graph from the list, a preview of this graph is displayed. All boxes in this preview image are labelled with constituents of the current name, which allows us to check, whether the components fit into the graph.

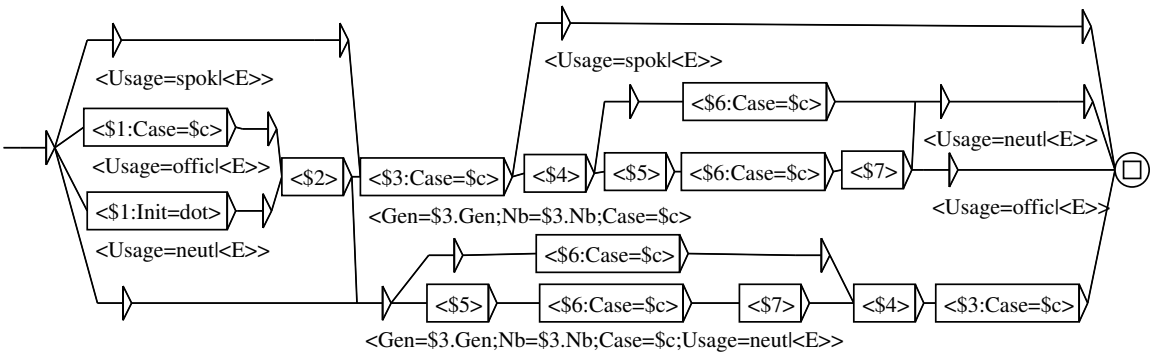

Fig. 7: Inflection graph for the name *Jan Rodowicz „Anoda”*

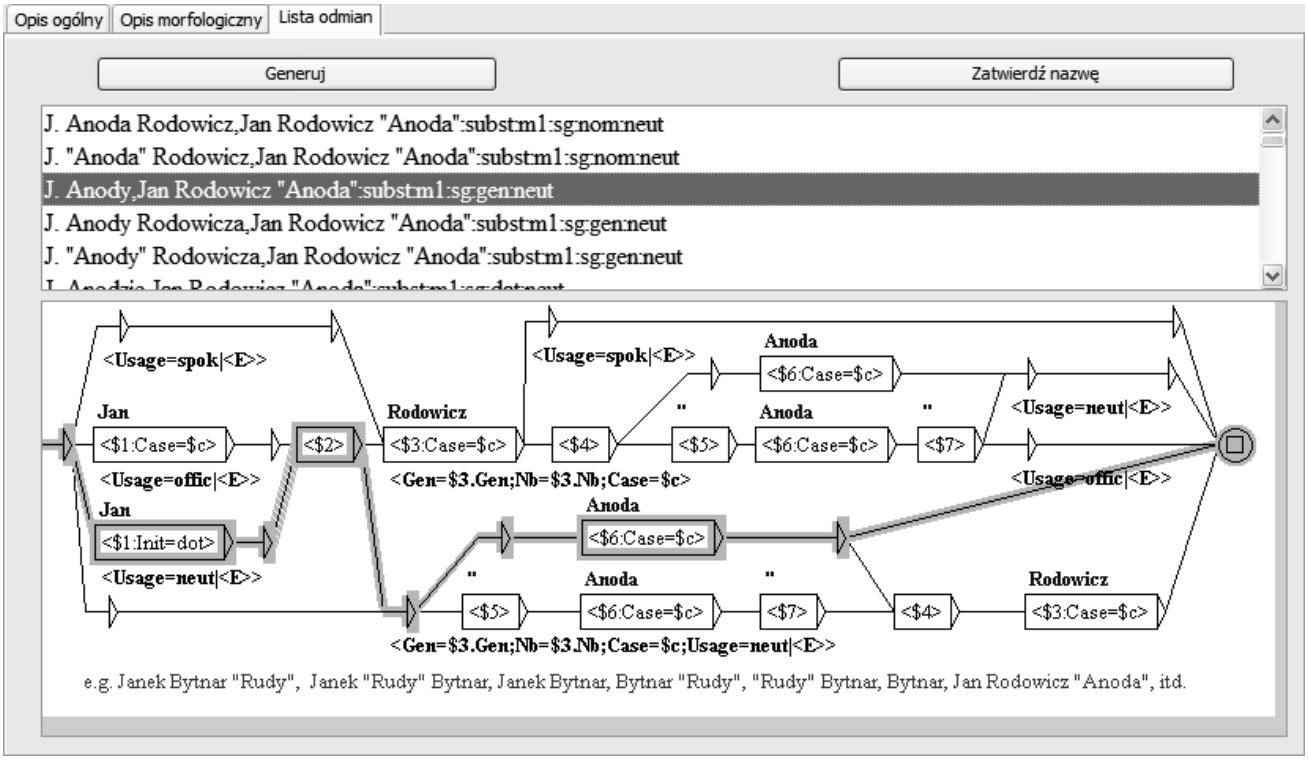

Fig. 8: The path corresponding to a given inflected form is highlighted in the graph preview

3.2. Tracing Paths in a Graph

For some names, even with the improvements of section 2.1., a high complexity of graphs is inevitable, as in Fig. 7 describing numerous combinations of a first name, a surname and a nickname.

After a graph is assigned to a name its inflection is validated by generating and checking all possible forms. If an erroneous form is generated, the lexicographer has to deduce, how this form was obtained. To simplify this task the new version of *Toposław* highlights the path in the graph corresponding to the form selected in the list of generated forms.

For example in Fig. 8 the path corresponding to the erroneous form *J. Anody* is highlighted (note the difference with the correct graph in Fig. 7). Automatic modification of the graph based on this information does not seem feasible, but when the path supporting the wrong form is identified, the graph can be corrected rather easily.

If a form can be generated by traversing more than one path, *Toposław* highlights all such paths. In the case of subcompounds only the path in the outermost graph is shown, as simultaneous visualisation of all levels would render the graph unreadable.

The implementation of this feature required changes both in *Multiflex* and in *Toposław*. *Multiflex* uses graphs in a compiled form (minimised and determinised), which has different paths than the graph defined by a lexicographer. To overcome this problem *Toposław*, in visualisation mode, produces a fake compiled form, which is very close to the original graph. This form is used by *Multiflex*, which for each generated form returns the path(s) that generated it. This information is subsequently used by *Toposław* to identify the respective boxes in the original graph.

3.3. New Graphs

Creating a new graph is speeded up, in that *Toposław* automatically generates a skeleton of the new graph consisting of boxes corresponding to the constituents of the compound. Since the lemma of the compound is always supposed to be present among possible forms, this „backbone" can be generated automatically.

4. Conclusions and Perspectives

We proposed several improvements of the *Multiflex* formalism, and graph managing and debugging in *Toposław*. As far as we know, visual debugging tools for inflection graphs are not available within other frameworks. The mechanism could be used for debugging graphs in the context of other *Unitex*-based applications (although some decoupling with *Multiflex* would be necessary). Our platform can support other languages whose morphological modules respect the interface constraints described in (Savary et al., 2009).

Further improvements are needed in handling numbers, years and dates frequent in urban names. Currently a separate graph is needed for each name containing such elements, as we need to represent the correspondence between the written and the spoken form e.g., 3 and *trzeci* 'third'. A general description of the equivalence between such forms would decrease the number and the complexity of graphs.

We are considering merging our methods with those proposed by (Abramowicz et al., 2006). This could give a good trade-off between the precision of hand-crafted rules and corpus-based automation, as well as between the appropriateness for recognition and generation.

5. References

Abramowicz, Witold, Agata Filipowska, Jakub Piskorski, Krzysztof Węcel, and Karol Wieloch, 2006. Linguistic Suite for Polish Cadastral System. In *Proceedings of LREC'06, Genoa, Italy*.

Krstev, Cvetana, Ranka Stanković, Duško Vitas, and Ivan Obradović, 2006. Workstation for Lexical Resources — Ws4LR. In *Proc. of LREC'06*.

Krstev, Cvetana and Duško Vitas, 2009. An effective method for developing a comprehensive morphological e-dictionary of compounds. In *Proceedings of Lexis and Grammar Conference, Bergen 2009*.

Marciniak, Małgorzata, Joanna Rabiega-Wiśniewska, Agata Savary, Marcin Woliński, and Celina Heliasz, 2009. Constructing an Electronic Dictionary of Polish Urban Proper Names. In *Recent Advances in Intelligent Information Systems*. Exit.

Paumier, Sébastien, 2003. Unitex 2.1. User manual. http://www-igm.univ-mlv.fr/∼unitex.

Piskorski, Jakub, Karol Wieloch, and Marcin Sydow, 2009. On knowledge-poor methods for person name matching and lemmatization for highly inflectional languages. *Information Retrieval*, 12(3):275–299.

Savary, Agata, 2005. A formalism for the computational morphology of multi-word units. *Archives of Control Sciences*, 15(3):437–449.

Savary, Agata, Joanna Rabiega-Wiśniewska, and Marcin Woliński, 2009. Inflection of Polish Multi-Word Proper Names with Morfeusz and Multiflex. *LNAI*, 5070.

Sikora, Piotr and Marcin Woliński, 2009. Toposław — a Dictionary Creation Tool. In *Recent Advances in Intelligent Information Systems*. Exit.

Woliński, Marcin, 2006. Morfeusz — a Practical Tool for the Morphological Analysis of Polish. In *Proc. of IIS: IIPWM'06*. Springer.