UNIVERSIDADE DA CORUÑA

LyS Research Group, Departamento de Computación

LyS

miopia
uuusa

Universal, Unsupervised, Uncovered Sentiment Analysis

User manual

David Vilares Calvo

Carlos Gómez Rodríguez

Miguel A. Alonso Pardo

June 20, 2016

Contents

1 Introduction

MIOPIA-UUUSA is a natural language processing (NLP) library for Universal, Unsupervised, Uncovered Sentiment Analysis (UUUSA). It is universal, because it provides you the capabilities for analysing the perception of the public with respect to a product, service, event or a celebrity, given a collection of related messages; **irrespective of the language, part-of-speech tagging or dependency parsing annotation guidelines.** It is (rule-based) unsupervised, because **it relies on subjectivity lexica** instead of labelled data with sentiment annotations. And it is uncovered, because **it can track and visualise how sentiment flows between words and phrases.**

This manual is a guide to describe how to install and exploit this software using the corresponding input formats. The reader should be familiar with concepts such as part-of-speech tagging, dependency parsing and sentiment analysis. Thus, this is a practical-oriented manual, where the theory of the model is not included. The latter can be found in:

- D. Vilares, C. Gómez-Rodríguez, and M. A. Alonso. Universal, Unsupervised, Uncovered Sentiment Analysis. *arXiv preprint arXiv:1606.05545*, 2016 [6].

MIOPIA-UUUSA is implemented as a Java library, named SAMULAN (Sentiment Analysis for Multiple LANguages). SAMULAN is a new library of MIOPIA, and in particular to its unsupervised syntax-based approach (part of MIOPIA-SD, software registration C-197-2015 by University of A Coruña, Spain), that was limited in terms of the language (only Spanish), training treebank (Ancora corpus) and the scope of the rules (intensification and negation) [5].

MIOPIA-UUUSA can be downloaded from `http://grupolys.org/software/UUUSA/`

MIOPIA-UUUSA has been also integrated at `http://miopia.grupolys.org`

2 Terms and conditions

You should have received a copy of the GNU General Public License along with MIOPIA-UUUSA. If not, see `http://www.gnu.org/licenses/`.

3 Requeriments

MIOPIA-UUUSA is a library for Sentiment Analysis purposes written in Java, so it can be used as long as the user has installed a Java Virtual Machine (version 8 or later).

3.1 Dependencies and resources

The system requirements and their dependencies can be found in the README.txt file, although they are also detailed below these lines:

- You should have installed a Java 8 (or later).
- You should have a semantic orientation lexicon (fine-grained) for your target language(s) (see §4.1).
- You should have defined a set of compositional operations (see §4.3).
- You should have a tagging model trained using Stanford part-of-speech tagger [2] and a parser trained with MaltParser [3]. Alternative, you might use a parsed CoNLL file (see §4.2 for further details) as the input file for MIOPIA-UUUSA, if you plan to use different tools for your natural language processing (NLP) pipeline.

3.2 Other issues

- CPU and memory requirements depends on the complexity of the tagging and the parsing models you are relying on to analyse your texts. In this respect, any relative modern machine able to run a Java Runtime Environment should be capable to run it.
- The length of your samples might also have impact in the memory size required by MIOPIA-UUUSA as well as in the execution time.
- MIOPIA-UUUSA is intended to work with different encodings. However, in order to avoid problems, we encourage to the users to employ *utf-8* encoding for all their files. If they present a different encoding, change it before starting to work with them.

4 The MIOPIA-UUUSA library

MIOPIA-UUUSA can be executed from command line (§4.5) or used as a part of your Java project (§4.6). It is implemented in Java in a packaged named SAMULAN. MIOPIA-UUUSA has not a graphical inferface as a part of its software, but it does have graphical demo available at `miopia.grupolys.org`.

4.1 SentiData

See a real example at `http://grupolys.org/software/UUUSA/`
SentiData is a directory structure to organise the files used by SentiStrength [4]. Among other files, it contains the minimum set of files required by MIOPIA-UUUSA to properly work. MIOPIA-UUUSA follows this same architecture, so existing SentiData for SentiStrength can be plugged into MIOPIA-UUUSA:

- *EmotionLookupTable.txt*: A fine-grained `tsv` file of 2 columns: (1) the subjective term and (2) its semantic orientation.
- *Emoticon_LookupTable.txt*: A fine-grained `tsv` file of 2 columns: (1) the emoticon form and (2) its semantic orientation [optional].
- *NegatingWordList.txt*: A fine-grained `tsv` file of 1 column containing the negating terms considered by the system [optional].
- *BoosterWordList.txt*: A fine-grained `tsv` file of 2 columns: (1) the intensifier term and (2) its intensification factor (where -1 implies that a subjective word becomes neutral) [optional].

It is possible to make some optional modifications to take advantage of the information provided by tools such as the part-of-speech tagger, for example:

- *X_EmoticonLookupTable.txt*: A emoticon lookup table for words belonging to a specific part-of-speech X (e.g. NOUN or ADJ). X must match a part-of-speech tag assigned by your tagger.
- *WordToLemmasStrippingList.txt*: A fine-grained `tsv` file of 2 columns: (1) the word and (2) a list of terminations, separated by semi-colons to be removed to guess its corresponding lemma lemma.

- *LemmasList.txt*: A fine-grained `tsv` file of 3 columns: (1) the part-of-speech tag, (2) the word and (3) its lemma form.

Additionally, MIOPIA-UUUSA Sentidata needs to contain non-regular files to work on raw input data:

- *.tagger file*: The tagging model trained wit the Stanford tagger.
- *maltparser*: A directory containing:
  - `.mco` file representing the trained parsing model.
  - `.xml` features file (see MaltParser documentation at `http://www.maltparser.org/` for detailed information) [optional].
  - `.conf` file. The output corresponging to the output `opt` file of the phase 3 of MaltOptimizer, renamed as a `.conf` file, in case you used MaltOptimizer to optimise the parameters of your MaltParser model [optional].

4.2 Parsed CoNLL files as input

See at real example at `http://grupolys.org/software/UUUSA/`
Instead of using raw input files, MIOPIA-UUUSA also allows to specify a parsed CoNLL file [1], tokenised, tagged and parsed with different NLP tools than the provided together with MIOPIA-UUUSA. Since one sample might have more than one dependency graph (specially when carrying out document-level sentiment classification), the file must contain a separated line of the form:

```
...
CoNLL graphs from sampleid-1 as in standard CoNLL file format
...
\n
### \t SAMPLEID \t CATEGORY\n
\n
...
CoNLL graphs from sampleid as in standard CoNLL file format
...
```

at the beginning of each sample, where `\t` is the tabulator character and `\n` represents the newline character.

4.3 Defining compositional operations

See a real example at `http://grupolys.org/software/UUUSA/`
We recommend to have a look to our paper [6] before creating new compositional operations. MIOPIA-UUUSA uses a `.xml` file to read the compositional operations defined by the user. Each rule is delimited by the `<rule>` field as showed in Example 1:

- `<forms>`: A set of word forms, separated by semicolons. It indicates a token that must match to trigger the operation. Alternative, the strings `SENTIDATA_NEGATION` and `SENTIDATA_BOOSTER` can be used to rely instead on the words contained in the NegatingWordList.txt and BoosterWordList.txt files, respectively. Regular expressions are also supported.
- `<dependency>`: A set of dependency types, separated by semicolons (e.g. subject,dobj). It indicates a dependency type that must match the source node to trigger the operation.
- `<postags>`: A set of part-of-speech tags, separated by semicolons (e.g. NOUN,ADJ). It indicates a part-of-speech tag that must match to trigger the operation.
- `<levelsup>`: An integer. It defines the number of levels from the node to spread before applying the operation to the target node/branch(es).
- `<rule>`: Defines the operation to be executed when the rule is triggered: `WEIGHTING` or `SHIFT`. Their format is as follow:
  - `WEIGHTING(`β`, scopes)`, where β is a real value indicating how to intensify the target scope as $(1+\beta) \times SO$, where `scopes` $\subseteq$ {`DEST,SUBJL,SUBJR,RCNX,LCNX, CHILDREN`} $\cup$ `D`, where `D` is the set of all valid dependency types. The selected scope must be separated by semicolons too (e.g. `DEST,dobj,CHILDREN`). `DEST` indicates that the scope of the operation is the SO of the node located at `levelsup` from the node that triggers the operation. `SUBJL` indicates that the scope of the operation is the semantic orientation of the first subjective left branch rooted at `levelsup` from the node that triggers the operation. `SUBJR` works similar to `SUBJL`, but applied to the first subjective right branch. `RCNX`, `LCNX` and `CHILDREN` are *backup* options: `RCNX` affects to the semantic orientation of the `X` closest children (e.g. `RCN2`) located on the right side of the node located `levelsup` from the node that

triggers the operation. `LCNX` works similar to `RCNX`, but considering the `X` closest children located on the left side. `CHILDREN` modifies the semantic orientation of the node rooted `levelsup` from the node that triggers the operation, once all operations to be applied on that level with a bigger priority have been computed. Alternative if `SENTIDA_BOOSTER` is used in the `<forms>` field, β can be replaced by the string `SENTIDATA`, to use the intensification assigned to the words occurring at the BoosterWordList.txt file.

  – `SHIFT(`α`,scopes)`, where α is the shifting real value represented as its absolute value. Usually, $\alpha = 4$.

- `<priority>`: An integer. It defines the priority of the operation when more than one operation needs to be applied over a target node (a larger number implies a bigger priority).
- `<validhead>` A set of part-of-speech tags (similar to the `<postags>` field). A filter to trigger the rule only if the part-of-speech tag of the head of the source node macthes this field. Set to *'*'* by default.

The special symbol *'*'* in one of the string fields represents that everything matches in that field. It is possible to find real `.xml` configuration files with real compositional operation at `http://grupolys.org/software/UUUSA/`.

Example 1. Example of a valid `xml` file composed of one compositional operation for the *'but'* clause. A new composed operation will be triggered when: the word form is *'but'*, its part-of-speech is *CONJ* and its dependency type is *cc*. It will affect (decreasing a 25% its semantic orientation) to the first left subjective children of the node that is the dependency head (that is, located 1 `levelsup`) of the *'but'* node.

```
<operations>
  <operation>
  <form>but</form>
  <postag>CONJ</postag>
  <dependency>cc</dependency>
  <type>WEIGHTING(-0.25,SUBJL)</type>
  <levelsup>1</levelsup>
  <priority>1</priority>
  <validhead>*</validhead>
  </operation>
</operations>
```

4.4 The properties file

See a real example at `http://grupolys.org/software/UUUSA/`

- *alwaysShift:* [true|false]. A shift operation always shifts, even if the semantic orientation of the scope is zero.
- *relatexEmotionSearch:* [true|false]. If a term does not appear in a specific part-of-speech emotion table it looks into the general EmotionLookupTable (if it exists).
- *joiner:* [default]. How to join the semantic orientation of different nodes. Default: simply sums.
- *binaryNeutralAsNegative:* [true|false]. If true and binary scale chosen, neutral messages are classified as negative. Otherwise they are classified as positive.
- *positiveWeighting:* The semantic orientation of positive terms is weighted as $(1 - positiveWeighting)$.
- *negativeWeighting:* The semantic orientation of negative terms is weighted as $(1 - negativeWeighting)$.

4.5 Running from command line

.

This section shows in detail how to run samulan from command line.

4.5.1 Command line options

- *input (–i)*: Specifies the path to the input raw data: a `tsv` file. It can have as many columns as wished as long as the text to analyse is in the last one.
- *sentidata (–s)*: Specifies the path to the SentiData, containing the data for your target language. This works similar to SentiStrength SentiData.
- *encoding (–e)*: Specifies the encoding of the input data. By default, `utf-8`.

- *rules (–r)*: Specifies the path to the `.xml` configuration file where the compositional operations are defined.
- *output (–o)*: Specifies the path to the output file. Using the standard output if not specified.
- *scale (–sc)*: : Specifies how to classify the output: `trinary`, `binary` or `so` (*semantic oriention*)]. `Trinary`: (1,positive), (0,neutral), (-1,negative). `Binary`: (1,positive), (-1,negative). `So`: a non normalised real value.
- *verbose (–v)*: [true] to show a verbose output. Omit otherwise.
- *properties (–p)*: Specifies the path to a samulan properties file. Explained in §4.4.
- *conll (–c)*: Specifies the path to a parsed CoNLL file. Explained in §4.2. Select instead of *input* when you plan to use a customised NLP pipeline.

Example 2. Command to run MIOPIA-UUUSA from command line to analyse raw data, given a sentidata directory named `EN-SentiData`, a `xml` file containing the rules named `configuration-EN.xml`, an input file named `en.txt` and a properties file names `samulan.properties`:

```
java -Dfile.encoding=UTF-8 -jar -Xmx2g samulan-0.1.0.jar \
-s EN-SentiData \
-r configuration-EN.xml \
-i en.txt \
-p samulan.properties \
-v true
```

This will analyse a text from the scratch (tokenisation, part-of-speech tagging, dependency parsing and sentiment analysis). Alternative, as explained previously, it is possible to use samulan to analyse already parsed files, which can be useful if you are running many experiments over the same dataset. This is showed in Example 3.

Example 3. Command to run MIOPIA-UUUSA from command line to analyse parsed CoNLL data, given a sentidata directory named `EN-SentiData`, a `xml` file containing the rules named `configuration-EN.xml` and an input file named `en_parsed.txt`, formatted as explained in §4.2:

```
java -Dfile.encoding=UTF-8 -jar -Xmx2g samulan-0.1.0.jar \
-s EN-SentiData \
-r configuration-EN.xml \
-c en_parsed.conll \
-p samulan.properties \
-v true
```

4.6 Using the Java library as a part of your project

MIOPIA-UUUSA is a `.jar` file and as any other `.jar` file it can be imported as a part of a Java Project. Check the Javadoc API provided together the project for more information.

5 Available examples and executables

At `http://grupolys.org/software/UUUSA/` demo explanations and examples are available.

6 Acknowledgements

This research is supported by the Ministerio de Economía y Competitividad (FFI2014-51978-C2). David Vilares is funded by the Ministerio de Educación, Cultura y Deporte (FPU13/01180). Carlos Gómez-Rodríguez is funded by an Oportunius program grant (Xunta de Galicia).

References

[1] S Buchholz and E Marsi. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, pages 149–164. Association for Computational Linguistics, 2006.

[2] M. De Marneffe and C. D. Manning. The Stanford typed dependencies representation. In *Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pages 1–8. Association for Computational Linguistics, 2008.

[3] J. Nivre, J. Hall, J. Nilsson, A. Chanev, G. Eryigit, S. Kübler, S. Marinov, and E. Marsi. MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2):95–135, 2007.

[4] M. Thelwall, K. Buckley, and G. Paltoglou. Sentiment strength detection for the social web. *J. Am. Soc. Inf. Sci. Technol.*, 63(1):163–173, 2012.

[5] D. Vilares, M. A. Alonso, and C. Gómez-Rodríguez. A syntactic approach for opinion mining on Spanish reviews. *Natural Language Engineering*, 21(1):139–163, 2015.

[6] D. Vilares, C. Gómez-Rodríguez, and M. A. Alonso. Universal, Unsupervised, Uncovered Sentiment Analysis. *arXiv preprint arXiv:1606.05545*, 2016.