**Natural Language Processing for PDF/TIFF/Image Documents   
Computer Vision for Image Data  
SYNTAX MODULE  
High Precision Natural Language Processing   
Technical Specification, Gap v0.91**

# 1 Words

## 1.1 Words Overview

The words NLP preprocessor contains the following primary classes, and their relationships:

* Words - This is the base class for the representation of a Natural Language Processing (NLP) preprocessed list of words. The constructor optionally takes as parameters the raw text to tokenize and flags for NLP preprocessing the text.

words = Words(“some text”, flags …)  
  
The constructor calls the private methods \_split(), \_stem() and \_stopwords().

* Word – A single NLP preprocessed word (token).
* Vocabulary – A performance optimized python dictionary for word classification and lemmatizing.

**Vocabulary**

**Words (Base Class)**

**Word**

**Word**

**Word**

**List**

**Fig. 1a High Level view of Words Class Object Relationships**

## 1.2 Words Initializer (Constructor)

***Synopsis***

Words( text, flags … )

***Parameters***

text : A Unicode string of text.

flags: Zero or more keyword parameters

bare : Tokenize only (True) - do not preprocess.  
  
 punct : Keep/classify (True) or remove (False) punctuation.  
  
 stopwords : Keep (True) all stop words or remove (False). If True, it supersedes all   
 other flags. If False, other flags may be used to include specific   
 categories. The stop words are a superset of the Porter list.  
  
 stem : Value indicating which stemmer to use:  
 - builtin: gap  
 - NLTK: porter, snowball, lancaster or the WordLemmatizer:   
 lemma  
  
 pos : Annotate(True) or not annotate (False) NLP preprocessed tokens with   
 parts of speech using NLTK pos\_tag().

spell : Spell check and replace misspelled words using pyaspeller.

roman : Romanize (True) or not Romanize (False) latin-1 encodings of NLP   
 preprocessed tokens into ASCII encoding.

number : Keep/classify (True) or remove (False) numerical numbers. Ex.  
 - 1 / 4.5 / 1,000 / three

unit : Keep/classify (True) or remove (False) units of measurement. Ex

* inches / ft / cm
* height / width / weight / ht / wt
* temperature / °F / °C

quantifier : Keep/classify (True) size specifying words or to remove (False). Ex.  
 - all / any / more

preposition : Keep/classify (True) preposition words or to remove (False). Ex.  
 - to / from / above

conjunction : Keep/classify (True) conjunction words or to remove (False). Ex.

* and / or / but

article : keep/classify (True) article words or to remove (False). Ex.  
 - a / an / the

demonstrative: Keep/classify (True) article words or to remove (False). Ex.  
 - this / that / these

question : Keep/classify (True)question words or to remove (False). Ex.  
 - who / want / how

pronoun : Keep/classify (True) pronoun words or to remove (False). Ex.  
 - he / she / them

date : Keep/classify/reduce (True) dates or to remove (False). Ex  
 - Jan. 1, 2000 / 01/01/2000 / 2000-01-01

dob : Keep/classify/reduce (True) date of births or to remove (False). Ex.  
 - DOB: Jan 1, 2000 / date of birth is 01-02-2012

ssn : Keep/classify (True) social security numbers or to remove (False). Ex.  
 - 123-84-1234 / 123 84 1234

telephone : Keep/classify (True) telephone numbers or to remove (False). Ex.  
 - (360) 123-1234 / +13601231234

name : Keep/classify (True) telephone numbers or to remove (False). Ex.

* Albert Einstein / Donald J. Trump

address : Keep/classify/reduce street address or to remove (False). Ex

* 124 NE 34th Cir, Home Town, AZ, 99123

gender : Keep/classify (True) gender specifying words or remove (False). Ex.

* male / man / gal / mom

sentiment : Keep/classify/reduce (True) sentiment word sequences or remove   
 (False).  
 - not bad / disgusting

***Exceptions***

A TypeError is raised if the parameter is not the expected type.

## 1.3 Words Properties

### 1.3.1 text

***Synopsis***

# Getter  
text = images.text

# Setter  
images.text = text

***Usage***

When used as a getter the property returns the original text.

When used as a setter the property re-preprocesses the text into machine learning ready data.

***Exceptions***

A TypeError is raised if the parameter is not the expected type.

### 1.3.2 bare *Synopsis*

# Getter  
tokens = images.bare

***Usage***

When used as a getter the property returns the NLP tokenized list unprocessed. All punctuation, words, capitalization and diacritic characters and script are preserved. The tokenized list is in a dictionary format of the form:

[ { ‘word’: word1, ‘tag’: tag }, {‘word’: word2, ‘tag’: tag } .. ]

Except for numbers and acronyms, the tag values are set to untagged (0).

### 1.3.3 Words

***Synopsis***

# Getter  
words = images.words

***Usage***

When used as a getter the property returns the NLP tokenized list according to the specified parameters. The tokenized list is in dictionary format of the form, when the parameter pos is False:

[ { ‘word’: word1, ‘tag’: tag }, {‘word’: word2, ‘tag’: tag } .. ]

Otherwise, when the pos parameter is set to True:

[ { ‘word’: word1, ‘tag’: tag, ‘pos’: POS }, {‘word’: word2, ‘tag’: tag, ‘pos’: POS } .. ]

### 1.3.4 bagOfWords

***Synopsis***

# Getter  
bag = images.bagOfWords

***Usage***

When used as a getter the property returns the word sequence as a Bag of Words, represented as a unordered dictionary, where the key is the word and the value is the number of occurrences:

{ ‘<word’> : <no. of occurrences>, … }

### 1.3.5 freqDist

***Synopsis***

# Getter  
freq = images.freqDist

***Usage***

When used as a getter the property returns the sorted tuples of a frequency distribution of words (from bag of words), in descending order (i.e., highest first)

[ ( ‘<word’>: <no. of occurrences> ), …. ]

### 1.3.5 termFreq

***Synopsis***

# Getter  
tf = images.termFreq

***Usage***

When used as a getter the property returns the sorted tuples of a term frequency distribution (percent that term occurs), in descending order (i.e., highest first)

[ ( ‘<word’>: <percentage of occurrences> ), …. ]

### 1.3.6 Static Variables

The Words class contains the following static variables:

* DECIMAL – The decimal point (US Standard: period , EU: comma)
* THOUSANDS – The thousandths unit separator (US Standard comma , EU period)

## 1.4 Words Overridden Operators

### 1.4.1 len()

***Synopsis***

nwords = len(words)

***Usage***

The len() (\_\_len\_\_) operator is overridden to return the number of NLP tokenized words.

### 1.4.2 +=

***Synopsis***

words += text

***Usage***

The += (\_\_iadd\_\_) method is overridden to add words to the sequenced word list (append).

## 1.5 Words Private Methods

The Words class contains the following private methods, which are called by the initializer:

* \_split() – This method performs the first phase of NLP preprocessing of the raw text into a sequenced list of words (bare processing mode).
  + Contractions are expanded (e.g., can’t => can not).
  + Newlines, carriage returns and tabs removed.
  + Duplicated whitespace is removed.
  + Text is split into words and punctuation.
  + Punctuation is removed (except in numerical and date representations when property number and/or date and/or dob is True).
* \_preprocess() - This method performs the second of NLP preprocessing of the ‘bare’ tokenized words.
  + Identify acronyms.
  + Identify proper names.
  + Words are lowercased.
  + Optionally words are Romanized, if roman attribute is set to True.
* \_stem() – This method performs the third phase of NLP preprocessing of the tokenized words by removing word endings and reducing word to its root stem (e.g., rider -> ride).
  + Remove plural endings (e.g., flies -> fly).
  + Remove past tense endings (e.g., baked -> bake).
  + Remove present participle endings (e.g., eating -> eat).
  + Remove verb to noun and comparative endings (e.g., rider -> ride, taller-> tall).
  + Remove noun to verb endings (e.g., flatten -> flat).
  + Remove adjective to adverb endings (e.g, costly -> cost).
  + Remove superlative endings (e.g., greatest -> great).
  + Spell check/replacement, if enabled, occurs prior to stemming.
* \_nltkStem() – This method uses the open source NLTK stemmer methods to perform the third phase of NLP preprocessing of the tokenized words, as an alternative to the internal stemmer (i.e. stem)). The Porter, Snowball, Lancaster and WordNetLemmatizer are selectable.
* \_stopwords() – This method performs the fourth phase of NLP preprocessing of the tokenized words by removing/keeping stop words.
  + Remove word (including infinity) and numeric representations of numbers, unless property number is True, then all numbers are retained.
    - If retained, EU decimal and thousandths unit separators converted to US standard.
    - +/- signs preserved.
    - Thousandths unit separator removed.
    - Hex numbers (starting with 0x prefix) are converted to integer value.
    - Text represented numbers (e.g., ten) are converted to integer value.
    - Text represented numeric ordering (e.g., 1st) are converted to integer value.
    - Fractions are converted to floating point value.
  + Remove units of measurement, unless property unit is True.
    - US Standard and Metric, including abbreviations, are recognized.
    - US and EU spelling of metric units are recognized.
  + Remove dates, unless property date is True.
  + Remove date of birth, unless property dob is True.
  + Remove USA social security numbers, unless property ssn is True, where the SSN number is converted to a 9 digit value.
  + Remove USA/CA telephone numbers, unless property telephone is True, where the telephone number is converted to a 10 digit number.
  + Remove USA/CA addresses, unless property address is True, where addresses are converted to the USPO addressing standard.
  + Remove gender indicating words (e.g., man) – inclusive of transgender, unless property telephone is True.
  + Remove proper names and titles (e.g., Dr.), unless property name is True.
  + Remove quantifier indicating words (e.g., all, any), unless property quantifier is True.
  + Remove prepositions (e.g., above, under), unless property preposition is True.
  + Remove conjunctions (e.g., and, or), unless property conjunction is True.
  + Remove articles (e.g., a, an), unless property article is True.
  + Remove demonstratives (e.g., this, that), unless property demonstrative is True.
  + Remove pronouns (e.g., his, her), unless property pronoun is True.
  + Remove question words (e.g., what, why), unless property question is True.
  + Remove common high frequency words i.e., Porter List).
  + Remove sentiment sequence (e.g., good, bad), unless sentiment property is True.
    - Sequence (e.g., not bad) replaced with “positive” or “negative”.
  + Remove punctuation and symbols, unless punct property is True.
* \_isdate() – This method is a support method for \_stopwords(). It will recognize date strings and convert them to ISO 8601 format. The following formats are recognized:
  + MM/DD/YY and MM/DD/YYYY
  + MM-DD-YY and MM-DD-YYYY
  + YYYY-MM-DD (ISO 8601)
  + Month Day, Year (e.g., January 6, 2016)
  + Abbr-Month Day, Year (e.g., Jan 6, 2016 and Jan. 6, 2016)

If the preceding word is birth or DOB, then the date will be tagged as a date of birth (vs. date).

* \_isnumber() - This method is a support method for \_stopwords(). It will recognize numerical sequences and convert them to decimal base 10 format. The following formats are recognized
  + Base 10 integer, floating point, exponent, fraction
  + Base 16 hex integers
* \_isSSN() - This method is a support method for \_stopwords(). It will recognize USA Social Security numbers. The following formats are recognized:
  + Prefixed with SSN or Soc. Sec. No. or Social Security Number
  + Number Format:
    - 12-123-1234 / 12 123 1234 / 121231234
* \_isTele() - This method is a support method for \_stopwords(). It will recognize USA/CA Telephone numbers. The following formats are recognized:
  + Prefixed with Tele, Phone, Mobile, Office, etc, optionally followed by Number, Num, No, #
  + Number Format:
    - 1231231234 / 123 123 1234 / 123-123-123 / (360) 123-1234 / …
* \_isAddr() – This method is a support method for \_stopwords(). It will recognize USA/CA postal addresses. The following formats are recognized:
  + [POB[,]] Street-Number [Street-Direction] Street-Name [Street-Type] [Street-Direction] [,] [POB[,]] [Secondary-Address[,]][City[,]State]
  + POB[,] City State
* \_streetnum() – This method supports the \_isAddr() method in recognizing street numbers. The following formats are recognized:
  + [(N|S|W|E)]digits[letter][-][digits|letter
  + Ex. N1300 / 123-33 / 33A / 33
* \_streetdir() – This method supports the \_isAddr() method in recognizing directional phrases. The following formats are recognized:
  + North|South[sp][West|East]
  + N|S[.][w|e][.]
* \_citystate() – This method supports the \_isAddr() method in recognizing city/state references in a postal address. The following formats are recognized:
  + City[,](Full-State|Abbr-State)

USA and Canadian state names are replaced with their ISO 3166-2 codes (e.g., Alabama => ISO3166-2:US-AL).

* \_pob() – This method supports the \_isAddr() method in recognizing USA and Canadian Post Office Boxes and Private Mail Boxes in street addresses. The following formats are recognized:
  + ( P.O.B | POB | P.O. Box | P.O. | PO ) digits [ (STN | RPO) words ]
  + ( P.M.B | PMB | P.M. Box ) digits
* \_streetaddr2() – This method supports the \_isAddr() method in recognizing secondary address components in street addresses. The following formats are recognized:
  + (Apt|Ste|Rm|Fl|Bldg|Dept)[#](word[-][word)
* \_postalcode() – This method supports the \_isAddr method() in recognizing USA and Canadian postal codes. The following formats are recognized:
  + 5digits[-4digits] # USA
  + [3letters][sp][3letters] # Canada
* \_isGender() – This method supports gender recognition in stopwords. It recognizes phrases:
  + (Sex|Gender)[:] (M|F|Male|Female)
* \_conversion() – This method performs the fifth phase of NLP preprocessing of the tokenized words of converting Standard to Metric (standard=True) and vice-versa (metric=True).
* \_partsofspeech() – This method performs the sixth phase of NLP preprocessing of the tokenized words of tagging words with their parts of speech tag (using NLTK).

## 1.5 Words Public Methods

The Words class contains no public methods.

## APPENDIX I: Updates

*Pre-Gap (Epipog) v1.1*

* Refactored Stopword Removal for Parts of Speech, Numbers and Dates
* Added Vocabulary class and Lemmatizing
* Added support for Date of Birth
* Added support for recognizing word version of numbers.
* Fixed handling of Hex numbers

*Pre-Gap (Epipog) v1.2*

* Added support for Social Security Numbers
* Added support for Telephone Numbers
* Added support for Proper Names
* Fix not recognizing Acronym if first word
* Refactored tokenization and added bare mode.

*Pre-Gap (Epipog) v1.3*

* Added support for Spanish punctuation: ¿¡
* Added support for numeric multipliers (e.g., 10 million).
* Added support for unit of measurements.
* Added conversion of unit of measurements between Standard and Metric.
* Added support for Sex/Gender[:] M/F
* Added support for USA street addresses
* Fix not recognizing single letter abbreviations.
* Fixed not recognizing title (name) proceeded by a comma.
* Fixed not recognizing number followed by unit of measurement when combined (e.g., 2cm).

*Pre-Gap (Epipog) v1.4*

* Added NLTK options for stemming, lemmatization and parts of speech.
* Change outputting of US State names to ISO 3166-2 standard.
* Added support for PMB in postal address.
* Added is/of separator between key and value (e.g., SSN is XXX-XX-XXXX).
* Added support for Canadian Street/Postal addresses.
* Added support for Romanizing latin-1 character encodings into ASCII.
* Added support for measurements

*Pre-Gap (Epipog) v1.5*

* Added Bag of Words, Word Frequency Distribution and Term Frequency (TF)

*Gap v0.91 (alpha)*

* Rewrote Specification.
* Added spell check/replacement
* Added UK to US spelling correction

## APPENDIX II: Anticipated Engineering

The following has been identified as enhancement/issues to be addressed in subsequent update:

1. Support for detecting Abbreviations
2. Support for email addresses
3. Add support for mail stop Secondary Street Address components.
4. Add support for Currency.
5. Add <SOS> and <EOS> annotation.
6. Fix lose next word after street/postal address

**Proprietary Information  
Copyright ©2018, Epipog, All Rights Reserved**