

preprocessing pipeline

```
def corpus2docs(corpus):
    fids = corpus.fileids()
    docs1 = []
    for fid in fids:
        doc_raw = corpus.raw(fid)
        doc = nltk.word_tokenize(doc_raw)
        docs1.append(doc)
    docs2 = [[w.lower() for w in doc] for doc in docs1]
    docs3 = [[w for w in doc if re.search('[a-z]+$', w)] for doc in docs2]
    docs4 = [[w for w in doc if w not in stop_list] for doc in docs3]
    docs5 = [[stemmer.stem(w) for w in doc] for doc in docs4]
    return docs5

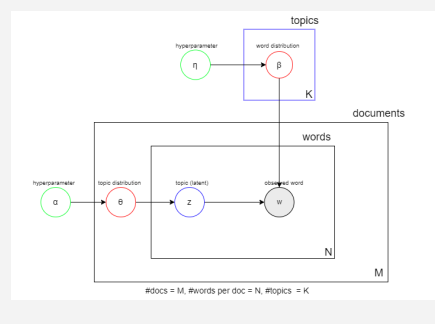
def docs2vecs(docs, dictionary):
    # docs is a list of documents returned by corpus2docs.
    # dictionary is a gensim.corpora.Dictionary object.
    vecs1 = [dictionary.doc2bow(doc) for doc in docs]
    tfidf = gensim.models.TfidfModel(vecs1)
    vecs2 = [tfidf[vec] for vec in vecs1]
    return vecs2
```

POS tags

N (noun)	dog, cat, chair
V (verb)	read, write, get
ADJ (adjective)	pretty, smart, blue
ADV (adverb)	gently, carefully, extremely
P (preposition)	in, on, by, with, about
PRO (pronoun)	I, me, mine, it, they...
CON (conjunction)	and, or, but, while, because
INT (interjection)	ooh, wow, yeah
DET (determiner)	all, his, they
AUX (auxiliary verb)	have done, might do
PAR (particle)	look up , get on
NUM (numeral)	one, two, three

Context-free grammar

LDA



LDA

- gibbs sampling**
1. random word-to-topic assignment
 2. re-assign each word to a topic, one by one, **assuming all other assignments are correct**

hyperparameters

- high α --> documents feature a mixture of most topics
- high β --> topics feature a mixture of most words

evaluation coherence (PMI), human eval

Sentiment-Topic Model (Plate Notation)

Discourse Markers

causal	because
consequence	as a result
conditional	if
temporal	when
additive	and
elaboration	[exemplification, rewording]
contrastive/concessive	but

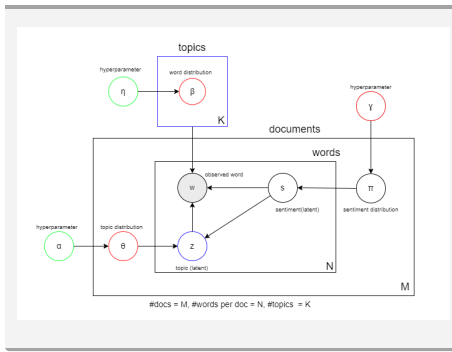
Preparation for NLTK classifier

```
#doc_tuple = (doc_representation, label)
> (('police':1, 'lawyer':1, 'courte')
#train_set = [doc_tuple1, doc_tuple2, ...]
```

```

Grammar = {
  objects: [
    Words/ tokens:
terminals,
    Right pos:
tags,
syntactic
tags,
sentence
  ];
  Rules: [
    X: node name, #eg
    " VP" (verb phrase)
    Y: sequence of
    objects that make up X #eg
    (V+NP)
  ]
}

```



Cluster Purity

$$P_i = \max_j P_{ij} \quad P_{ij} = \frac{\#docs(class = j, cluster = i)}{\#docs(cluster = i)}$$

Overall purity

$$\sum_{i=1}^K \left[\frac{\#docs(cluster = i)}{\#docs} \cdot P_i \right] \quad K = \#clusters$$

Cluster Entropy

$$e_i = - \sum_{j=1}^J P_{ij} \log_2 P_{ij}$$

Pointwise Mutual Information

$$PMI(w_i, w_j) = \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)}$$

Morphemes

stems, affixes (prefix/suffix). Useful for POS tagging and text normalization

Semantics

synonyms	diff words, same meaning
polyseme	same word, diff meaning
hypernym/-hyponym	category >>> specific
meronym/m-etonym	part >>> whole