1. ***Features***
   1. **Feature of word【词特征】（词本身的特征，wordshape=normal）**
      1. useWord：词向量
      2. usePrev：使用前一个单词的特征，如果useTags为True，则加入前一个单词的tag
      3. useNext：Gives you feature for (nw,c), and together with other options enables other next features, such as (nt,c) [with useTags)
      4. useWordPairs：Features for (pw, w, c) and (w, nw, c)
   2. **Tags【part of speech】（使用词的Tag做特征，即使用pos tag）**
      1. useTags：使用单词的tag，这里的tag应该是指pos tag
      2. tags特征也可以配合usePrev，useNext和useWordPairs使用
   3. **n-gram【N元文法模型】（第n个词出现的概率仅依赖于他前面出现的n-1个词，n=1是unigram，n=2是bigram，n=3是trigram）（n-gram有两种应用，一种是判断一个句子的合理性，这种方式得到的是一个概率；一种是计算两个字符串的相似度，在模糊匹配中常会用到）**
      1. useNGrams：假设有一个字符串 s，那么该字符串的N-Gram就按长度 N 切分原词得到的词段，也就是 s 中所有长度为 N 的子字符串。这里用到的特征是从一个word（letter）的n-gram中提取特征（提取方法？）
      2. noMidNGrams：如果word的n-gram不包括首或尾的字符，则不使用这些n-gram的特征
      3. lowercaseNGrams：从只有小写的n-gram中提取特征
      4. dehyphenateNGrams：在计算n-gram之前去掉字符串中的连字符，如在一个单词换行时为了连接行末和行首而加入的横杠
      5. useNeighborNGrams：使用前一个词和当前词的n-grams，对于汉语比较有用
      6. maxNGramLeng：If this number is positive, n-grams above this size will not be used in the model， Default is -1
      7. cacheNGrams：If true, record the NGram features that correspond to a String (under the current option settings) and reuse rather than recalculating if the String is seen again.
         1. 当corpus数量很大的时候，该方法应该可以帮助算法节省运行时间
   4. **gazette：We call a gazette a mapping between a list of tokens and an entity kind. If that list of tokens matches exactly on your text, then that would be tagged as an entity.**
      1. Class + entry : class是指某个entity的label，entry可以是一个token，也可以是一个phrase，使用gazette是一种基于规则识别NER的方法，如果定义了某个label，那么在训练集中所有满足entry的token都会被标做gazette中指定的label
      2. 把token或者phrase对应到某一个entity，生成一个dictionary。对与dictionary有两种用法：
         1. 直接exact match或者partial match，直接使用label，可以提高分类效率
         2. 使用dictionary的特征作为训练特征
            1. <http://idf.github.io/assets/pdf/Ingredient_Unit_and_Amount_Named_Entity_Recognition_and_Relationship_Identification_in_Cooking_Speech_Transcripts.pdf>
            2. <https://openproceedings.org/2017/conf/edbt/paper-257.pdf>
         3. 在stanford的NER中:  It simply provides another feature for the CRF to train against. If the CRF has higher weights for other features, the gazette features may be overwhelmed.
         4. 在Stanford的CRF模型中，如何设置feature的权重值？
      3. Gazette的另一个用法是让词组（phrase）对应到一个label，但是这种情况效果不如使用stanford的regexner or the tokensregex tools included in Stanford CoreNLP.
      4. useGazettes：使用gazettes feature
         1. 在测试集或者预测的时候如何使用Gazettes的feature的？
   5. **Class feature【类别特征】**
      1. useClassFeature：Include a feature for the class (as a class marginal). Puts a prior on the classes which is equivalent to how often the feature appeared in the training data.
      2. maxLeft：the maximum context of class features used.
      3. maxRight：The number of things to the right that have to be cached to run the Viterbi algorithm: the maximum context of class features used. The maximum possible clique size to use is (maxLeft + maxRight + 1)
      4. useSequences： 这里的class都是指类别， 也就是label，label的特征？class combination features
      5. usePrevSequences：Does not use any class combination features using previous classes if this is false
      6. useNextSequences：Does not use any class combination features using next classes if this is false
   6. **wordshape features【词型特征】**
      1. wordShape：通过不同的计算shape的公式可获得不同的特征，shape包括很多，比如是否存在大小写，是否数字等等
         1. 很多种wordShape的计算方式，如wordShape=chris2useLC
         2. 在WordShapeClassifier.java中可以看到很多种计算wordshape的方法，这里在计算WordShape是，首先遍历一个词的每个字符，如果这个字符是一个数字，则把这个位置上的字符替换成’d’，如果是小写字母换成’x’，大写字符换成’X’，这样下来，如果一个词是’HelloWorld’就会变成’XxxxxXxxxx’()
      2. useTypeSeqs：Use basic zeroeth order word shape features.
      3. useTypeSeqs2：Add additional first and second order word shape features
      4. useTypeSeqs3：Adds one more first order shape sequence
   7. **Disjunctive【析取特征】**
      1. useDisjunctive：【离散数学】析取
      2. disjunctionWidth：The number of words on each side of the current word that are included in the disjunction features，Default is 4.
   8. **Lemmas Feature【词根特征】**
      1. useLemmas：Include the lemma of a word as a feature.
      2. usePrevNextLemmas：Include the previous/next lemma of a word as a feature.
      3. useLemmasAsWord：Include the lemma of a word as a feature.
   9. **Normalize【归一化，标准化，预处理】**
      1. normalizeTerms：If this is true, some words are normalized: day and month names are lowercased (as for normalizeTimex) and some British spellings are mapped to American English spellings (e.g., -our/-or, etc.).
      2. nomalizeTimeX：If this is true, capitalization of day and month names is normalized to lowercase
   10. **Print【打印信息】**
       1. printFeatures：print out all the features generated by the classifier for a dataset to a file based on this name (starting with "features-", suffixed "-1" and "-2" for train and test). This simply prints the feature names, one per line.
       2. When training a model, there will produce a file named features-1.txt which contains all train data features, but when testing a model, the features-1.txt will be overwrite by the test data features.
       3. printClassifier：Style in which to print the classifier. One of: HighWeight, HighMagnitude, Collection, AllWeights, WeightHistogram
   11. **Clique【是一种特征提取的结构】**
       1. Clique表示一个数组，其中0表示当前位置，-1表示前一个位置，1表示后一个位置，在提取特征的时候，总是以Clique为单位生成一个FeatureList
       2. CliqueC={0}，Return featuresC
       3. CliqueCpC={-1,0}，Return featuresCpC
       4. CliqueCp2C={-2,0}，Return featuresCp2C
       5. CliqueCp3C
       6. CliqueCp4C
       7. CliqueCp5C
       8. CliqueCpCp2C={-2，-1,0}，Return featuresCpCp2C
       9. CliqueCpCp2Cp3C
       10. CliqueCpCp2Cp3Cp4C
       11. CliqueCpCp2Cp3Cp4Cp5C
       12. CliqueCnC={0,1}， Return featuresCnC
       13. CliqueCpCnC={-1,0,1}，Return featuresCpCnC
   12. **Annotations**
       1. Return String.Class
       2. Annotation是提取特征的方式，如果要新加入一个特征，需要定义一个不重复的特征名，然后将特征提取的方式通过新的annotation加入到FeatureFactory中
       3. Processing is done using a bag of features model, with all of the features mixed together, which is why it is important to not have any name conflicts
2. ***How to generate Features（Bag of Features Model）***
   1. BOF（图像检索）
   2. 首先，我们用surf算法生成图像库中每幅图的特征点及描述符。

再用k-means算法对图像库中的特征点进行训练，生成类心。

生成每幅图像的BOF，具体方法为：判断图像的每个特征点与哪个类心最近，最近则放入该类心，最后将生成一列频数表，即初步的无权BOF。

通过tf-idf对频数表加上权重，生成最终的bof。（因为每个类心对图像的影响不同。比如超市里条形码中的第一位总是6，它对辨别产品毫无作用，因此权重要减小）。

对query进来的图像也进行3.4步操作，生成一列query图的BOF。

将query的Bof向量与图像库中每幅图的Bof向量求夹角，夹角最小的即为匹配对象。

1. ***Using（+号的数量反映了特征的显著性）***
   1. useTags ++
   2. usePrev +
   3. useNext ++
   4. useWordPairs +
   5. usePrev,useSequences,usePrevSequences +++
   6. useDisjunctive(p) +++
   7. useDisjunctive(n) ++++
   8. useNGrams (noMidNGrams, MaxNGramLeng, lowercaseNGrams, dehyphenateNGrams) +++
   9. wordShape >= 0 +++
   10. wordShape >= 0,useTypeSeqs,useTypeSeqs2,maxLeft>=2 +++

***3.Demo（例子中使用的特征组合，也是重点研究的特征组合）***

useClassFeature=true

useWord=true

useNGrams=true

noMidNGrams=true

maxNGramLeng=6

usePrev=true

useNext=true

useSequences=true

usePrevSequences=true

maxLeft=1

useTypeSeqs=true

useTypeSeqs2=true

useTypeySequences=true

wordShape=chris2useLC

useDisjunctive=true