SOLUTION NOTES FOR EXAMINERS AND SUPERVISORS ONLY

Information Retrieval 2005 – Paper 7 Question 12 (SHT)

This SNOWBALL system is discussed in Lecture 6. Question a) is bookwork. Question b) makes the students think through a concrete example, something that was not done in lectures but that should not be too hard. Questions c) and d) are less obvious and require some thinking.

The SNOWBALL algorithm uses bootstrapping from known tuples of named entities which stand in a well-defined relationship, in order to detect new tuples.

(*a*) **Describe SNOWBALL's algorithm in detail, including the thresholds used in the single steps of the algorithm. [7 marks]**

- *Hypothesise patterns*: Occurrences of the established patterns within a window of X words are collected. A vector space representation of the patterns is compiled. Weights concerning the left, right and middle contexts are set (thresholds).
- *Cluster patterns*: Similar patterns are clustered together; a similarity threshold is applied (threshold A). The distance of each pattern from the centroid pattern is calculated ($MatchC_i, P_i$), which will later serve as weight.
- *Evaluate patterns:* All patterns occurring more often than a threshold (B) are considered. For evaluation, all occurrences in the text are considered where ORGANISATION is known. The accuracy of each pattern is estimated as the number of times the right combination is found by the pattern (*P.positive*), by the number of times the pattern is found in text (*P.positive* + *P.negative*).

  $$Conf(i) = \frac{P.positive}{P.positive + P.negative}$$

  .

  Productivity is also taken into account:

  $$Conf_{R \log FNorm}(P) = \frac{Conf(P) \log_2(P.positive)}{max_{i \in P} Conf(i)}$$

  Throw away patterns under a pattern confidence threshold (C).
- *Hypothesise tuples*: Find all tuples that occur with the new patterns, and calculate to which degree they match the generalised pattern ($MatchC_i, P_i$).
- *Evaluate tuples*: Confidence of a tuple T is the probability that at least one valid tuple is produced:

  $$Conf(T) = 1 - \prod_{i=0}^{|P|} (1 - Conf(P_i) \cdot Match(C_i, P_i)$$

  Throw away tuples under a tuple confidence threshold (D).

- Reset confidence of patterns; threshold w_{updt} (E) controls learning rate (does system trust new occurrences?).

(*b*) **The table below contains corpus examples of co-occurrences of organisation names (o) and location names (l). Consider a situation where SNOWBALL is applied to the corpus examples given here, when the only known tuples are <Microsoft, Redmond> and <Exxon, Irving>.**

A	**<l>Seattle<l>-based company <o>Boeing</o> offered...**
B	**Yesterday, at <o>Microsoft</o> ’s headquarters in <l>Redmond<l> the deal was brokered...**
C	**Though they had never been at <l>Redmond<l>, <o>Microsoft<o> showed them**
D	**In <l>New York</l>, <o>Microsoft<o> stock nosedived**
E	**When we arrived in <l>London<l>, <o>Exxon<o> petrol stations were...**
F	**... met at <o>Microsoft<o> headquarters. In <l>Redmond<l>,**
G	**<l>Boeing<l>, <o>Seattle<o>, had no choice but to ...**
H	**In <l>New York</l>, <o>Intel<o> stock recovered**
I	**...due to arrive in <l>Irving<l>,<o>Exxon<o>-shared might**
J	**<o>Boeing<o> headquarters in <l>Seattle<l>**
K	**<o>Microsoft<o>, <l>Redmond<l>, made a statement ...**
L	**<o>Boeing<o>, <l>Seattle<l>, confirmed ...**
M	**<o>Microsoft<o>, <l>Redmond<l>, readily agreed ...**
N	**... <o>Exxon<o>. Although they had never in their whole life been in <l>Irving<l>, they ...**
O	**<o>Exxon</o>, <l>New York<l>, was a winner in our recent...**

Discuss which patterns get hypothesised and which new tuples this produces in the next iteration. Assume sensible thresholds. [6 marks]

- The following patterns get hypothesised because of the known tuples:

  1. “at” LOCATION “’s(.5) headquarters(1.0) .(.5) In(.5) in(.5)” ORGANISATION – observed twice (B, F).
  2. ORGANISATION “,(1.0)” LOCATION “,(1.0)” – observed three times (K, M, O).
  3. LOCATION “,” ORGANISATION – observed twice (C,I).
  4. ORGANISATION “.(1.0) Although(1.0) they(1.0) had(1.0) never(1.0) in(1.0) their(1.0) whole(1.0) life(1.0) been(1.0) in(1.0) LOCATION – observed once (N).

- If threshold $\tau_{sup} = 2$, pattern 4 will be eliminated. Pattern 1 gets confirmed with Conf(P_2) $= \frac{2}{2}$; productivity will add factor $\log_2$. Pattern 3 gets rejected because of $\frac{|C,I|}{|C,D,E,I|}$ will probably be below pattern confidence threshold and get thrown out. Pattern 2 gets a reasonably high Conf of $\frac{2}{3}$. Taking productivity into account will add in $\log_2$.

- Tuple <Boeing,Seattle> gets added due to J (Pattern 1), and to G, which is associated with Pattern 2 which got a lower confidence value. (We can assume for here that all patterns are equally far from generalised SNOWBALL pattern).
- In the next iteration, the pattern from example A will get evaluated.

(*c*) **What happens to the result in part (*b*) if the sentence "*Microsoft's previous headquarters in Cincinnati were insured for 20 million dollars.*" gets added to the corpus? [3 marks]**

There are two options: either this context does not get classified into pattern 1. This would be the preferable option. Whether or not this would happen depends on the similarity metric, the dimensionality of the features used for clustering, and on how many examples occur. If it does not get clustered with Pattern 1, it will be discarded, which is good, because it is not a pattern indicating headquarter-type contexts. If it does however get clustered with pattern 1, it will lower the confidence of pattern 1 and thus the confidence of the tuple <Boeing, Seattle> – though its higher distance from the centroid than the good contexts B and F will make sure that the contribution of this context is lower than that of the other patterns, which means that the tuple <Boeing, Seattle> should still get in.

(*d*) **The SNOWBALL algorithm is to be applied to find tuples of person names and their professional positions from a large newspaper corpus. Would you expect SNOWBALL to work well on this task, and why? [4 marks]**

SNOWBALL should work less well on this task. Firstly, these contexts require three named entities: name of company, name of person, job description (such as CEO). This will make patterns combinatorically explode, and the system might run into data sparseness problems. Secondly, the time factor also makes this task harder. People move in and out of important jobs, and within a corpus there will be several mentions of people in a certain position, which is valid only for a certain time frame. Thus, the evaluation of the patterns, which hinges on the existence of a key by which one can automatically assess correctness, will work much less well. The key/uniqueness constraints work because each company has only one headquarter, and because these don't change often.