Worker Metrics (in <job_id>_workerMetrics.xlsx)

Number of annotations per sentence: is the average number of different relations used by a worker for annotating a set of sentences.

Worker agreement: is measure of the coincidence between a worker annotations and the annotations of the rest of the workers. Annotations "coincidence" is the number of relations in common between the annotators of a sentence.

The agreement value of a particular worker is defined as the aggregated pairwise agreement between the worker and the rest of workers, weighted by the number of sentences in common.

In more detail:

$$agreement(wi) = jiagr(wi,wj)jinumSentencesInCommon(wi,wj)$$

Where the agreement between two workers is defined as:

$$agr(wi,wj) = numSentencesInCommon(wi,wj)$$
$$kS'relationsInCommon(wi,wj,Sk)kS' numAnnotations(wi,Sk)$$

Where

- S' is the subset of sentences in common between w_i and w_j,
- relationsInCommon(w_i, w_j, S_k) is the number of relation annotations in common between the annotations of w_i and w_j for the sentence S_k.
- numAnnotations(w_i, S_k) is the number of relations w_i has used for annotating the sentence S_k.

Cosine similarity: is a measure of similarity between the annotations of a worker and the aggregated annotations of the rest.

Cosine similarity express the degree of similarity between two vectors, obtained by calculating the cosine of the angle between them. The similarity between the worker's annotation and the aggregated annotation (subtracting the worker's vector) is first computed, by calculating the cosine of both vectors. This reflects how "close" the relation(s) chosen by the worker is to the opinion of the majority for that sentence.

This values are then averaged for all the sentences, thus measuring the average similarity of the workers annotations with those of the rest, for all the sentences of a job.

Worker-sentence score: is a measure of the quality of the annotations of a worker for a sentence. It is defined, for each sentence and worker, as the difference between the *Sentence Clarity* for the sentence minus the *worker cosine similarity* for that sentence. It provides a measure of how "good" the annotation of the worker is in relation to the sentence clarity.

A low worker-sentence score reflects that the cosine similarity value is close to the sentence clarity value, implying that the worker annotation is similar to that of the majority. This holds true regardless of how unanimous that majority is: when the sentence clarity value is low -scattered vector-, it is hard to reach an agreement on the relation; therefore -even though the cosine score will be low- it's more likely to be close to the sentence clarity value. Thus, in spite of having a low cosine agreement value, the worker-sentence score will be good.

On the other hand, when the sentence clarity is high but the cosine similarity for the worker is low, that implies the worker annotation is very different from that of the majority, hence the annotation will be considered low-quality.

How is implemented:

For each sentence the worker has annotated:

1. Calculate the cosine(restVector, workerVector)
2. $Score(worker_j, sent_i) = cosine(sent_i, worker_j) - sentenceClarity(sent_i)$

Where:
sentenceVector is the vector containing the aggregated annotations of all workers for a sentence and *restVector* = *sentenceVector* - *workerVector*

Worker-relation score: is measure of the quality of the annotations of a worker for a particular relation. It is defined, for each relation and worker, as the difference between the *Relation Clarity* minus the average *worker-sentence score* of the worker for the sentences that have been annotated mostly with that relation.

The worker-sentence score measures the quality of the annotations for a sentence; by averaging it over the sentences for which the relation has been identified, an estimation of the quality of the workers annotation for that particular relation is obtained. As in the worker-sentence score, this is agreement value is subtracted from the Relation clarity score, to obtain a score that takes into account the relation clarity for evaluating the workers annotations.

A high worker-relation score would be obtained combining high relation clarity values and low worker-sentence scores (little distance to the majority's opinion), reflecting overall quality of the worker annotations for the relation is good.

How is implemented: (in measures.R -> workerRelationScore)

1. For or each of the sentences in the dataset, the relation that annotated by the majority is. The sentences are grouped by the relation labelled. S[R] is the group of sentences that the majority of annotators have labelled with the relation R.
2. For each worker and relation, the subset $S_\omega[R]$ of sentences of S[R] that the worker ω has annotated is extracted.
3. For all the sentences in $S_\omega[R]$, the average value of the cosine between the worker Vectors and the sentence Vectors is computed.

   [1] $avg(cos(S'_\omega[R], W_\omega))$

   Where W_ω is the set of worker annotations for the sentences in $S_\omega[R]$ and $S'_\omega[R]$ is the set of "RestVectors": each of aggregated sentence vectors substracting the worker ω annotation for that sentence.
1. Worker-Relation score [ω, R] = RelationClarity(R) - $avg(cos(S'_\omega[R], W_\omega))$

Sentence metrics (in <job_id>_sentenceMetrics.xlsx -> sentenceMetrics)

Sentence-relation score: is the core crowd truth metric for relation extraction. It is measured for each relation on each sentence as the cosine of the unit vector for the relation with the sentence vector. The relation score is used for training and evaluation of the relation extraction system, it is viewed as the probability that the sentence expresses the relation. This is a fundamental shift from the traditional approach, in which sentences are simply labelled as expressing, or not, the relation, and presents new challenges for the evaluation metric and especially for training.

How is implemented: (in measures.R -> sentRelationScore)

The unit vector for a relation is the vector of null values for all relations, except one.
For instance, the unit vector for the relation "Symptom" for the sentence 123456 is:

```
        D  S C M L AW P SE IA PO T CI OTH NONE
123456 0  1  0  0 0    0 0    0 0    0 0 0      0         0
```

For each sentence S:
    For each relation R:
        sentRelScore[S,R] = cosine(sentenceVector[S], unitVector[R])

Sentence clarity: is defined for each sentence as the max relation score for that sentence. If all the workers selected the same relation for a sentence, the max relation score will be 1, indicating a clear sentence. Sentence clarity is used to weight sentences in training and evaluation of the relation extraction system, since annotators have a hard time classifying them, the machine should not be penalized as much for getting it wrong in evaluation, nor should it treat such training examples as exemplars.

The implementation is trivial: it returns the max value for the sentence-relation score vector (see the previous section for its implementation):

Sentence Clarity(S) = max (sentRelScore[S,])

Relation metrics (in <job_id>_sentenceMetrics.xlsx -> relationMetrics)

Relation similarity: is a pairwise conditional probability that if relation R_i is annotated in a sentence, relation R_j is as well. Information about relation similarity is used in training and evaluation, as it roughly indicates how confusable the linguistic expression of two relations are. This would indicate, for example, that relation co-learning would not work for similar relations.

$$\text{RelationSimilarity}(A,B) = P(A \mid B) = P(A\ B)P(B)$$

$$P(A) = \text{numOccurrences}(A)\text{totalNumberOfRelations}$$

$$P(A \cap B) = \text{numOccurrences}(A \cap B)\text{totalNumberOfRelations}$$

NOTE: P(A| B) != P(B | A).

- numOccurrences(A): number of times the relation A has been used in the job, both as a single relation and in combination with other relations.

- numOccurrences(A ∩ B): number of times the relations A and B are used in **the same annotation** by a worker. Other relations may also be part of that annotation.

$$\text{totalNumberOfRelations} = i\ \ Sj\ W\text{numRelations}(si,wj)$$

Where numRelations(s_i,w_j) is the number of relations the worker $w_j \in W$ has used for annotating the sentence $s_i \in S$.

How to read the table (Relation Similarity, in sentenceMetrics.xlsx -> “relation-metrics” tab) :
Table[A, B] = P(B | A).

	D	S	C	M	L	AW	P

D	0	P(S | D)	P(C | D)	P(M | D)	P(L | D)	P(AW | D)	P(P | D)
S	P(D | S)	0	P(C | S)	P(M | S)	P(L | S)	P(AW | S)	P(P | S)
C	P(D | C)	P(S | C)	0	P(M | C)	P(L | C)	P(AW | C)	P(P | C)

How to read P(A | B): is “the probability that A is annotated in a sentence, given that B is annotated”. Or the probability that if the relation B is annotated in a sentence, relation A is annotated as well.

Looking at the table, the probability in Table[D,S] = probability of the relation S is annotated, given that the relation D is annotated. If “S” and “D” are similar, it’s more likely that both relations appear in the aggregated sentence vector. Or the other way around: the higher possibility of two relations appearing together in the sentence vector, the more likely is that both relations are similar.

In general, in the row “D”, the probabilities for a relation being present as well, given that the relation “D” is already on the sentence vector. Similarly, the Column “D” shows the probability of the presence of “D” in a sentence vector, considering the other relations that are part of the vector.

Difference between P(A | B) and P(B | A):

$P(A \mid B) = P(B \cap A) / P(B)$
$P(B \mid A) = P(A \cap B) / P(A)$

If $P(A) > P(B)$, then $P(A \mid B) > P(B \mid A)$.

This is equivalent to say that the conditional probability of A given B is bigger if the probability of A is bigger by itself, before accounting for event B. In other words, if A is more likely to be used than B, then the probability of A given B will be higher than the probability of B given A [Because $P(B \cap A) = P(A \cap B)$]

Relation ambiguity: is defined for each relation as the max relation similarity for the relation. If a relation is very clear, then it will have a low score. Since techniques like relation co-learning have proven effective, it may be a useful property of a set of relations to exclude ambiguous relations from the set.

How is implemented: by taking the max of each column in the relation similarity score matrix.

Relation Ambiguity(R) = Max(Column(R))

Relation clarity : is defined for each relation as the max sentence-relation score for the relation over all sentences. If a relation has a high clarity score, it means that it is at least possible to express the relation clearly. We find in our experiments that a lot of relations that exist in structured sources a very difficult to express clearly in language, and are not frequently present in textual sources. Unclear relations may indicate unattainable learning tasks.

How is implemented:

Given the Sentence Relation score (see before):

relationClarity(R) = max(sentRelScore[,R])

(= max of the relation column of the matrix).