Entailment: An Effective Metric for Comparing and Evaluating Hierarchical and Non-hierarchical Annotation Schemes

Rohan Ramanath*
R. V. College of Engineering, India
ronramanath@gmail.com

Monojit Choudhury **Kalika Bali**
Microsoft Research Lab India
{monojitc, kalikab}@microsoft.com

Abstract

Hierarchical or *nested* annotation of linguistic data often co-exists with simpler non-hierarchical or *flat* counterparts, a classic example being that of annotations used for parsing and chunking. In this work, we propose a general strategy for comparing across these two schemes of annotation using the concept of *entailment* that formalizes a correspondence between them. We use crowdsourcing to obtain query and sentence chunking and show that entailment can not only be used as an effective evaluation metric to assess the quality of annotations, but it can also be employed to filter out noisy annotations.

1 Introduction

Linguistic annotations at all levels of linguistic organization – phonological, morpho-syntactic, semantic, discourse and pragmatic, are often hierarchical or *nested* in nature. For instance, syntactic dependencies are annotated as *phrase structure* or *dependency trees* (Jurafsky and Martin, 2000). Nevertheless, the inherent cognitive load associated with nested segmentation and the sufficiency of simpler annotation schemes for building NLP applications have often lead researchers to define non-hierarchical or *flat* annotation schemes. The flat annotation, in essence, is a "flattened" version of the tree. For instance, *chunking* of Natural Language (NL) text, which is often considered an essential preprocessing step for many NLP applications (Abney, 1991; Abney, 1995), is, loosely speaking, a flattened version of the phrase structure tree. The closely related task of *Query Segmentation* is of special interest to us here, as it is the first step in further analysis and understanding of Web search queries (Hagen et al., 2011).

f	Pipe representation	*Boundary var.*
3	barbie dress up | games	0 0 1
3	barbie dress | up games	0 1 0
2	barbie | dress up | games	1 0 1
2	barbie | dress up games	1 0 0

Table 1: Example of flat segmentations from 10 Turkers. f is the frequency of annotations; segment boundaries are represented by |.

The task in both query and sentence chunking is to divide the string of words into contiguous substrings of words (commonly refered to as *segments* or *chunks*) such that the words from a segment are related to each other more strongly than words from different segments. It is typically assumed that the segments are syntactically and semantically coherent. Table 1 illustrates the concept of segmentation of a query. The crowdsourced annotations for this data were obtained from 10 annotators, the experimental details of which will be described in Sec. 5. We shall refer to this style of text chunking as *flat segmentation*.

Nested segmentation of a query or a sentence, on the other hand, is a recursive application of flat segmentation, whereby the longer flat segments are further divided into smaller chunks recursively. The process stops when a segment consists of less than three words or is a multiword entity that cannot be segmented further. This style of segmentation can be represented through nested parenthesization of the text, as illustrated in Table 2. These annotations were also obtained through the same crowdsourcing experiment (Sec. 5). Fig. 1 shows an alternative visualization of a nested segmentation in the form of a tree.

An important problem that arises in the context of flat segmentation is the issue of granular-

*The work was done during author's internship at Microsoft Research Lab India.

Proceedings of the 7th Linguistic Annotation Workshop & Interoperability with Discourse, pages 42–50,
Sofia, Bulgaria, August 8-9, 2013.

f	Bracket representation	*Boundary var.*
4	((barbie dress)(up games))	0 1 0
3	(barbie ((dress up) games))	2 0 1
2	(barbie (dress (up games)))	2 1 0
1	((barbie (dress up)) games)	1 0 2

Table 2: Example of nested segmentation from 10 Turkers. f is the frequency of annotations.

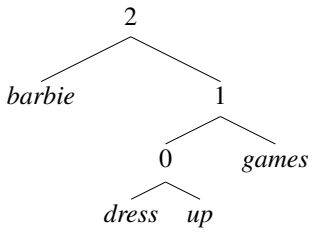

Figure 1: Tree representation of the nested segmentation: (*barbie* ((*dress up*) *games*))

ity. For instance, in the case of NL chunking, it is not clear whether the chunk boundaries should correspond to the innermost parentheses in the nested segmentation marking very short chunks, or should one annotate the larger chunks corresponding to clausal boundaries. For this reason, Inter-Annotator Agreement (IAA) for flat annotation tasks is often poor (Bali et al., 2009; Hagen et al., 2011; Saha Roy et al., 2012). However, low IAA does not necessarily imply low quality annotation, and could as well be due to the inherent ambiguity in the task definition with respect to granularity. Although we have illustrated the concept and problems of flat and nested annotations using the examples of sentence and query segmentation, these issues are generic and typical of any flat annotation scheme which tries to flatten or approximate an underlying hierarchical structure. There are three important research questions pertaining to the *linguistic annotations* of this kind:

- How to measure the true IAA and the quality of the flat annotations?
- How to compare the agreement between the flat and the nested annotations?
- How can we identify or construct the optimal or error-free flat annotations from a noisy mixture of nested and flat annotations?

In this paper, we introduce the concept of "*entailment* of a flat annotation by a nested annotation". For a given linguistic unit (a query or a sentence, for example), a nested annotation is said to *entail* a flat annotation if the structure of the latter does not contradict the more specific structure represented by the former. Based on this simple notion, which will be formalized in Sec. 3, we develop effective techniques for comparing across and evaluating the quality of flat and nested annotations, and identifying the optimal flat annotation. We validate our theoretical framework on the tasks of query and sentence segmentation. In particular, we conduct crowdsourcing based flat and nested segmentation experiments for Web search queries and sentences using Amazon Mechanical Turk (AMT)[1]. We also obtain annotations for the same datasets by trained experts which are expected to be of better quality than the AMT-based annotations. Various statistical analyses of the annotated data bring out the effectiveness of *entailment* as a metric for comparison and evaluation of flat and nested annotations.

The rest of the paper is organized as follows. Sec. 2 provides some background on the annotation tasks and related work on IAA. In Sec. 3, we introduce the notion of entailment and develop theoretical models and related strategies for assessing the quality of annotation. In Sec. 4, we introduce some strategies based on entailment for the identification of error-free annotations from a given set of noisy annotations. Sec. 5 describes the annotation experiments and results. Sec. 6 concludes the paper by summarizing the work and discussing future research directions. All the annotated datasets used in this research can be obtained freely from `http://research.microsoft.com/apps/pubs/default.aspx?id=192002` and used for non-commercial research purposes.

2 Background

Segmentation or chunking of NL text is a well-studied problem. Abney (1991; 1992; 1995) defines a chunk as a sub-tree within a syntactic phrase structure tree corresponding to Noun, Prepositional, Adjectival, Adverbial and Verb Phrases. Similarly, Bharati et al (1995) define it as Noun Group and Verb Group based only on local surface information. Chunking is an important preprocessing step towards parsing.

Like chunking, query segmentation is an important step towards query understanding and is generally believed to be useful for Web search

[1] `https://www.mturk.com/mturk/welcome`

(see Hagen et al. (2011) for a survey). Automatic query segmentation algorithms are typically evaluated against a small set of human-annotated queries (Bergsma and Wang, 2007). The reported low IAA for such datasets casts serious doubts on the reliability of annotation and the performance of the algorithms evaluated on them (Hagen et al., 2011; Saha Roy et al., 2012). To address the issue of data scarcity, Hagen et al. (2011) created a large set of manually segmented queries through crowdsourcing[2]. However, their approach has certain limitations because the crowd is already provided with a few possible segmentations of a query to choose from. Nevertheless, if large scale data has to be procured crowdsourcing seems to be the only efficient and effective model for the task, and has been proven to be so for other IR and linguistic annotations (see Lease et al. (2011) for examples). It should be noted that almost all the work on query segmentation, except (Huang et al., 2010), has considered only flat segments.

An important problem that arises in the context of *flat* annotations is the issue of granularity. In the absence of a set of guidelines that explicitly state the granularity expected, Inter-Annotator Agreement (IAA) for flat annotation tasks are often poor. Bali et al. (2009) showed that for NL chunking, annotators typically agree on major (i.e., clausal) boundaries but do not agree on minor (i.e., phrasal or intra-phrasal) boundaries. Similarly, for query segmentation, low IAA remains an issue (Hagen et al., 2011; Saha Roy et al., 2012).

The issue of granularity is effectively addressed in *nested* annotation, because the annotator is expected to mark the most atomic segments (such as named entities and multiword expressions) and then recursively combine them to obtain larger segments. Certain amount of ambiguity, that may arise because of lack of specific guidelines on the number of valid segments at the last level (i.e., topmost level of the nested segmentation tree), can also be resolved by forcing the annotator to recursively divide the sentence/query always into exactly two parts (Abney, 1992; Bali et al., 2009).

The present study is an extension of our recent work (Ramanath et al., 2013) on analysis of the effectiveness of crowdsourcing for query and sentence segmentation. We introduced a novel IAA metric based on Kripendorff's α, and showed that while the apparent agreement between the annotators in a crowdsourced experiment might be high, the chance corrected agreement is actually low for both flat and nested segmentations (as compared to gold annotations obtained from three experts). The reason for the apparently high agreement is due to an inherent bias of the crowd to divide a piece of text in roughly two equal parts. The present study extends this work by introducing a metric to compare across flat and nested segmentations that enables us to further analyze the reliability of the crowdsourced annotations. This metric is then employed to identify the optimal flat segmentation(s) from a set of noisy annotations. The study uses the same experimental setup and annotated datasets as described in (Ramanath et al., 2013). Nevertheless, for the sake of readability and self-containedness, the relevant details will be mentioned here again.

We do not know of any previous work that compares flat and nested schemes of annotation. In fact, Artstein and Poesio (2008), in a detailed survey of IAA metrics and their usage in NLP, mention that defining IAA metrics for trees (hierarchical annotations) is a difficult problem due to the existence of overlapping annotations. Vadas and Curran (2011) and Brants (2000) discuss measuring IAA of nested segmentations employing the concepts of precision, recall, and f-score. However, neither of these studies apply statistical correction for chance agreement.

3 Entailment: Definition and Modeling

In this section, we shall introduce certain notations and use them to formalize the notion of entailment, which in turn, is used for the computation of agreement between flat and nested segmentations. Although we shall develop the whole framework in the context of queries, it is applicable to sentence segmentation and, in fact, more generally to any flat and nested annotations.

3.1 Basic Definitions

Let Q be the set of all queries. A query $q \in Q$ can be represented as a sequence of $|q|$ words: $w_1 w_2 \ldots w_{|q|}$. We introduce $|q| - 1$ random variables, $b_1, b_2, \ldots b_{|q|-1}$, such that b_i represents the boundary between the words w_i and w_{i+1}. A flat and nested segmentation of q, represented by F_q^j and N_q^j respectively, j varying from 1 to total number of annotations, $\mathbf{c}$, is a particular instantiation of these boundary variables as follows.

[2] http://www.webis.de/research/corpora

Definition. *Flat Segmentation*: A flat segmentation, F_q^j, can be uniquely defined by a binary assignment of the boundary variables b_i^j, where $b_i^j = 1$ iff w_i and w_{i+1} belong to two different flat segments. Otherwise, $b_i^j = 0$. Thus, q has $2^{|q|-1}$ possible flat segmentations.

Definition. *Nested Segmentation*: A nested segmentation, N_q^j, is defined as an assignment of non-negative integers to the boundary variables such that $b_i^j = 0$ iff words w_i and w_{i+1} form an atomic segment (i.e., they are grouped together), else $b_i^j = 1 + max(left_i, right_i)$, where $left_i$ and $right_i$ are the heights of the largest subtrees ending at w_i and beginning at w_{i+1} respectively.

This numbering scheme can be understood through Fig. 1. Every internal node of the binary tree corresponding to the nested segmentation is numbered according to its height. The lowest internal nodes, both of whose children are query words, are assigned a value of 0. Other internal nodes get a value of one greater than the height of its higher child. Since every internal node corresponds to a boundary, we assign the height of the node to the corresponding boundary variables. The number of unique nested segmentations of q is the corresponding *Catalan number*[3] $C_{|q|-1}$.

Note that, following Abney's (1992) suggestion for nested chunking, we define nested segmentation as a strict binary tree or binary bracketing of the query. This is not only helpful for theoretical analysis, but also necessary to ensure that there is no ambiguity related to the granularity of segments.

3.2 Entailment

Given a nested segmentation N_q^j, there are several possible ways to "flatten" it. Flat segmentations of q, where $b_i = 0$ for all i (i.e., the whole query is one segment) and $b_i = 1$ for all i (i.e., all words are in different segments) are trivially obtainable from N_q^j, and therefore, are not neither informative nor interesting. Intuitively, any flat segmentation, F_q^k, can be said to agree with N_q^j if for every flat segment in F_q^k there is a corresponding internal node in N_q^j, such that the subgraph rooted at that node *spans* (contains) all and only those words present in the flat segment (Abney, 1991).

Let us take the examples of flat and nested segmentations shown in Tables 1 and 2 to illustrate this notion. Consider two nested segmentations, $N_q^1 = ((barbie\ (dress\ up))\ games)$, $N_q^2 = (barbie\ ((dress\ up)\ games))$ and three flat segmentations, $F_q^1 = barbie \mid dress\ up \mid games$, $F_q^2 = barbie \mid dress\ up\ games$, $F_q^3 = barbie\ dress \mid up\ games$. Figure 2 diagrammatically compares the two nested segmentations (the two rows) with the three flat segmentations (columns A, B and C). There are three flat segments in F_q^1, of which the two single word segments $barbie$ and $games$ trivially coincide with the corresponding leaf nodes. The segment $dressup$ coincides exactly with the words spanned by the node marked 0 of N_q^1 (Fig. 2, top row, column A). Hence, F_q^1 can be said to be in agreement with N_q^1. On the other hand, there is no node in N_q^1, which exactly coincides with the segment $dressupgames$ of F_q^2 (Fig. 2, top row, column B). Hence, we say that N_q^1 does not agree with F_q^2.

We formalize this notion of agreement in terms of *entailment*, which is defined as follows.

Definition: *Entailment.* A nested segmentation, N_q^j is said to *entail* a flat segmentation, F_q^k, (or equivalently, F_q^k is *entailed by* N_q^j) if and only if for every multiword segment $w_{i+1}, w_{i+2}, ..., w_{i+l}$ in F_q^k, the corresponding boundary variables in N_q^j follows the constraint: $b_i > b_{i+m}$ and $b_{i+l} > b_{i+m}$ for all $1 \leq m < l$.

It can be proved that this definition of entailment is equivalent to the intuitive description provided earlier. Yet another equivalent definition of entailment is presented in the form of Algorithm 1. Due to paucity of space, the proofs of equivalence are omitted.

Definition: *Average Observed Entailment.* For the set of queries Q, and corresponding sets of $\mathbf{c}$ flat and nested segmentations, there are $|Q|\mathbf{c}^2$ pairs of flat and nested segmentations that can be compared for entailment. We define the *average observed entailment* for this annotation set as the fraction of these $|Q|\mathbf{c}^2$ annotation pairs for which the flat segmentation is entailed by the corresponding nested segmentation. We shall express this fraction as percentage.

3.3 Entailment by Random Chance

Average observed Entailment can be considered as a measure of the IAA, and hence, an indicator of the quality of the annotations. However, in order to interpret the significance of this value, we need an estimate of the average entailment that

[3] http://en.wikipedia.org/wiki/Catalan_number

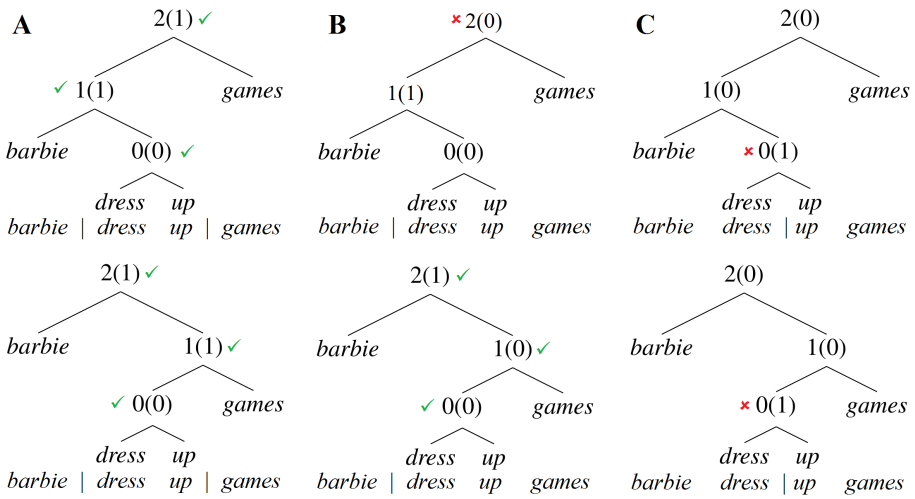

Figure 2: Every node of the tree represent boundary values, nested(flat). Column A: F_q^1 is entailed by both N_q^1 and N_q^2, Column B: F_q^2 is entailed by N_q^2 but not N_q^1, Column C: F_q^3 is entailed by neither N_q^1 nor N_q^2. The nodes (or equivalently the boundaries) violating the entailment constraint are marked a cross, and those agreeing are marked with ticks.

Algorithm 1 Algorithm: isEntail

procedure ISENTAIL($flat, nested$) ▷ flat, nested are lists containing boundary values
  if $len(nested) \leq 1$ or $len(flat) \leq 1$ **then**
    return True
  end if
  $h \leftarrow$ largest element in $nested$
  $i \leftarrow$ index of h
  if $flat[i] = 1$ **then**
    if ! $isEntail(flat[:i], nested[:i])$ or ! $isEntail(flat[i+1:], nested[i+1:])$ **then**
      return False
    else
      return True
    end if
  else
    while $h \neq 0$ **do**
      $nested[i] \leftarrow -nested[i]$
      $h \leftarrow$ largest element in $nested$
      $i \leftarrow$ index of h
      if $flat[i] = 1$ **then**
        return False
      end if
    end while
    return True
  end if
end procedure

one would expect if the annotations, both flat and nested, were drawn uniformly at random from the set of all possible annotations. From our experiments we observe that trivial flat segmentations are, in fact, extremely rare, and a very large fraction of the flat annotations have two or three segments. Therefore, for computing the chance entailment, we assume that the number of segments in the flat segmentation is known and fixed, which is either 2 or 3, but all segmentations with these many segments are equally likely to be chosen. We also assume that all nested segmentations are equally likely.

When there are 2 segments: For a query q, the number of flat segmentations with two segments, i.e., one boundary, is $\binom{|q|-1}{1} = |q| - 1$. Note that for any nested segmentation N_q^j, all flat segmentations that have at least one boundary and is entailed by it must have a boundary between w_{i^*} and w_{i^*+1}, where b_{i^*} has the highest value in N_q^j. In other words, b_{i^*} is the boundary corresponding to the root of the nested tree (the proof is intuitive and is omitted). Therefore, there is *exactly one* "flat segmentation with one boundary" that is entailed by a given N_q^j. Therefore, the random chance that a nested segmentation N_q^j will entail a flat segmentation with one boundary is given by $(|q| - 1)^{-1}$ (for $|q| > 1$).

When there are 3 segments: Number of flat segmentations with two boundaries is $\binom{|q|-1}{2}$. The flat segmentation(s) entailed by N_q^j can be generated as follows. As argued above, every flat segmentation entailed by N_q^j must have a boundary

at position i^*. The second boundary can be either in the left or right of i^*. But in either case, the choice of the boundary is unique which will correspond to the highest node in the left or right subtree of the root node. Thus, every nested segmentation entails at most 2 flat segmentations. However, if $i^* = 1$ or $|q| - 1$ for a N_q^j, then, respectively, the left or right subtrees do not exist. In such cases, there is only one flat segmentation entailed by N_q^j. Note that there are exactly $C_{|q|-2}$ nested segmentations for which the $i^* = 1$, and similarly another $C_{|q|-2}$ for which $i^* = |q| - 1$. Therefore, out of $C_{|q|-1} \times \binom{|q|-1}{2}$ pairs, exactly $2C_{|q|-1} - 2C_{|q|-2}$ pairs satisfy the entailment conditions. Thus, the expected probability of entailment by random chance when there are exactly two boundaries in the flat segmentation of q is:

$$\frac{2(C_{|q|-1} - C_{|q|-2})}{C_{|q|-1}\binom{|q|-1}{2}} = 2\binom{|q|-1}{2}^{-1}(1 - \frac{C_{|q|-2}}{C_{|q|-1}})$$

The values of the probability of observing a random nested segmentation entailing a flat segmentation with exactly two boundaries for $|q| = 3, 4, 5, 6, 7$ and 8 are 1, 0.4, 0.213. 0.133, 0.091 and 0.049 respectively.

3.4 Other IAA Metrics

Although entailment can be used as a measure of agreement between flat and nested segmentations, IAA within flat or within nested segmentations cannot be computed using this notion. In (Ramanath et al., 2013), we have extensively dealt with the issue of computing IAA for these cases. Krippendorff's α (Krippendorff, 2004), which is an extremely versatile agreement coefficient, has been appropriately modified to be applicable to a crowdsourced annotation scenario. $\alpha = 1$ implies perfect agreement, $\alpha = 0$ implies that the observed agreement is just as good as that by random chance, whereas $\alpha < 0$ implies that the observed agreement is less than that one would expect by random chance. Due to paucity of space we omit any further discussion on this and refer the reader to (Ramanath et al., 2013). Here, we will use the α values as an alternative indicator of IAA and therefore, the quality of annotation.

4 Optimal Segmentation

Suppose that we have a large number of flat and nested annotations coming from a noisy source such as crowdsourcing; is it possible to employ the notion of entailment to identify the annotations which are most likely to be correct? Here, we describe two such strategies to obtain the optimal (error-free) flat segmentation.

Flat Entailed by Most Nested (FEMN): The intuition behind this approach is that if a flat segmentation F_q^k is entailed by most of the nested segmentations of q, then it is very likely that F_q^k is correct. Therefore, for each flat segmentations of q, we count the number of nested segmentations of q that entail it, and the one with highest count is declared as the optimal FEMN segmentation. It is interesting to note that while computing the optimal FEMN segmentation, we never encountered a tie between two flat segmentations. The trivial flat segmentations (i.e., if the whole query is one segment or every word is in different segments) are filtered as a preprocessing step.

Iterative Voting (IV): FEMN assumes that the nested segmentations are relatively noise-free. If most of the nested segmentations are erroneous, FEMN would select an erroneous optimal flat segmentation. To circumvent this issue, we propose a more sophisticated *iterative voting* process, where we count the number of flat segmentations entailed by each nested segmentation of q, and similarly, number of nested segmentations that entail each flat segmentation. The flat and nested segmentations with the least scores are then removed from the dataset. Then we recursively apply the IV process on the reduced set of annotations until we are left with a single flat segmentation.

5 Experiments and Results

We obtained nested and flat segmentation of Web search queries through crowdsourcing as well as from trained experts. Furthermore, we also conducted similar crowdsourcing experiments for NL sentences, which helped us understand the specific challenges in annotating queries because of their apparent lack of a well-defined syntactic structure.

In this section, we first describe the experimental setup and datasets, and then present the observations and results.

5.1 Crowdsourcing Experiment

In this study we use the same set of crowdsourced annotations as described in (Ramanath et al., 2013). For the sake of completeness, we briefly describe the annotation procedure here as

well. We used Amazon Mechanical Turk for the crowdsourcing experiments. Two separate Human Intelligence Tasks were designed for flat and nested segmentation. The concept of flat and nested segmentation was introduced to the Turkers with the help of two short videos[4].

When in doubt regarding the meaning of a query, the Turkers were advised to issue the query on a search engine of their choice and find out its possible interpretation(s). Only Turkers who had completed more than 100 tasks at an acceptance rate of $\geq 60\%$ were allowed to participate in the task and were paid \$0.02 for a flat and \$0.06 for a nested segmentation. Every query was annotated by 10 different annotators.

5.2 Dataset

The following sets of queries and sentences were used for annotations:

Q500, QG500: Saha Roy et al. (2012) released a dataset of 500 queries, 5 to 8 words long, for the evaluation of various segmentation algorithms. This dataset has flat segmentations from three annotators obtained under controlled experimental settings, and could be considered as *Gold* annotation. Hence, we selected this set for our experiments as well. We procured the corresponding nested segmentation for these queries from two human experts who are regular search engine users. They annotated the data under supervision and were trained and paid for the task. We shall refer to the set of flat and nested gold annotations as **QG500**, whereas **Q500** will be reserved for the dataset procured through the AMT experiments.

Q700: As 500 queries are not enough for making reliable conclusions and also, since the queries may not have been chosen specifically for the purpose of annotation experiments, we expanded the set with another 700 queries sampled from the logs of a popular commercial search engine. We picked, uniformly at random, queries that were 4 to 8 words long.

S300: We randomly selected 300 English sentences from a collection of full texts of public domain books[5] that were 5 to 15 words long, and manually checked them for well-formedness.

[4]Flat: `http://youtu.be/eMeLjJIvIh0`, Nested: `http://youtu.be/xE3rwANbFvU`

[5]`http://www.gutenberg.org`

5.3 Entailment Statistics

Table 3 reports two statistics – the values of Kripendorff's α and the average observed entailment (expressed as %) for flat and nested segmentations along with the corresponding expected values for entailment by chance. For nested segmentation, the α values were computed for two different distance metrics[6] d_1 and d_2.

As expected, the highest value of α for both flat and nested segmentation is observed for the gold annotations. An $\alpha > 0.6$ indicates a *reasonably good*[7] IAA, and thus, reliable annotations. We note that the entailment statistics follow a very similar trend as α, and for all the cases, the observed average entailment is much higher than what we would expect by random chance. These two observations clearly point to the fact that entailment is indeed a good indicator of the agreement between the nested and flat segmentations, and consequently, the reliability of the annotations. We also observe that the average entailment for **S300** is in the same ballpark as for the queries. This indicates that the apparent lack of structure in queries does not specifically influence the annotations. Along the same lines, one can also argue that the length of a text, which is higher for sentences than queries, does not affect the crowdsourced annotations. In fact, in our previous study (Ramanath et al., 2013), we show that it is the bias of the Turkers to divide a text in approximately two segments of equal size (irrespective of other factors, like syntactic structure or length), that leads to very similar IAA across different types of texts. Our current study on entailment further strengthens this fact.

Figure 3 plots the distribution of the entailment values for the three datasets. The distributions are normal-like implying that entailment is a robust metric and its average value is a usable statistic.

In order to analyze the agreement between the Turkers and the experts, we computed the average entailment between **Q500** flat annotations (from AMT) with **QG500** nested annotations, and similarly, **Q500** nested annotations with **QG500**

[6]Intuitively, for d_1 disagreements between segment boundaries are equally penalized at all the levels of nested tree, whereas for d_2 disagreements higher up the tree (i.e., close to the root) are penalized more than those at lower levels.

[7]It should be noted that there is no consensus on what is a good value of α for linguistic annotations, partly because it is dependent on the nature of the annotation task and the demand of the end applications that use the annotated data.

Dataset	Krippendorff's α			Entailment Statistics	
	Flat	*Nested*		*Observed*	*Chance*
	d_1	d_1	d_2		
Q700	0.21	0.21	0.16	49.68	12.63
Q500	0.22	0.15	0.15	56.69	19.08
QG500	0.61	0.66	0.67	87.07	11.91
S300	0.27	0.18	0.14	52.86	19.12

Table 3: α and Average Entailment Statistics

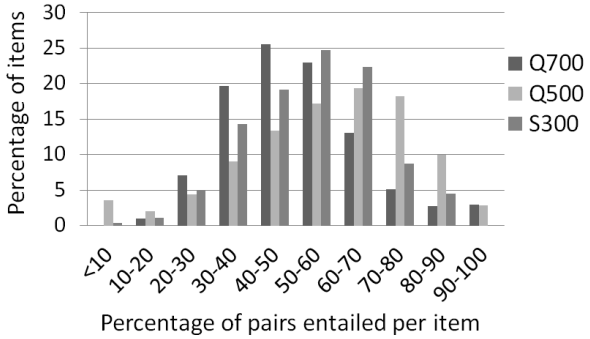

Figure 3: Distribution of the entailment values (x-axis) plotted as the % of comparable flat-nested annotation pairs.

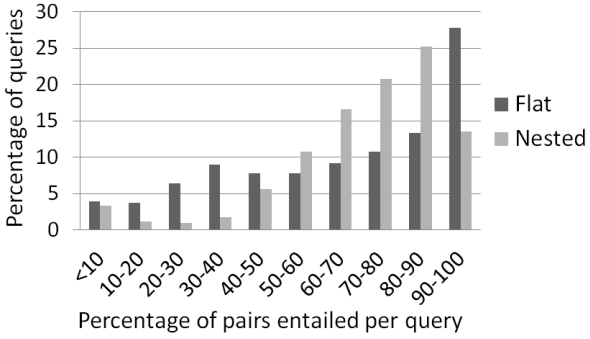

Figure 4: Distribution of percentage of entailed pairs using **QG500** as reference.

flat annotations, which turned out to be 70.42% and 63.24% respectively. The corresponding distributions are shown as *Nested* and *Flat* in Fig. 4. Thus, the flat segmentations from the Turkers seem to be more accurate than their nested segmentations, a fact also supported by the α values. This could be due to the much higher cognitive load associated with nested segmentation that demands more time and concentration that an ordinary Turker may not be willing to invest.

5.4 Optimal Segmentation Results

In order to evaluate the optimal flat segmentation selection strategies, FEMN and IV, we computed the percentage of queries in **Q500** for which the optimal flat segmentation (as obtained by applying these strategies on AMT annotations) is entailed by the corresponding nested segmentations in **QG500**. The average entailment values for FEMN and IV turns out to be 79.60% and 82.80% respectively. This shows that the strategies are indeed able to pull out the more accurate flat segmentations from the set, though, as one would expect, IV performs better than FEMN, and its chosen segmentations are almost as good as that by expert annotators.

Another experiment was conducted to precisely characterize the effectiveness of these strategies whereby we mixed the annotations from the **Q500** and **QG500**, and then applied FEMN and IV to pull out the optimal flat segmentations. We observed that for 63.71% and 91.44% of the queries, the optimal segmentation chosen by FEMN and IV respectively was indeed one of the three gold flat annotations in **QG500**. This reinforces our conclusion that IV can effectively identify the optimal flat segmentation of a query from a noisy set of flat and nested segmentations.

6 Conclusion

In this paper, we proposed entailment as a theoretical model for comparing hierarchical and non-hierarchical annotations. We present a formalization of the notion of entailment and use it for devising two strategies, FEMN and IV, for identifying the optimal flat segmentation in a noisy set of annotations. One of the main contributions of this work resides in our following experimental finding: Even though annotations obtained through crowdsourcing for a difficult task like query segmentation might be very noisy, a small fraction of the annotations are nevertheless correct; it is possible to filter out these correct annotations using the Iterative Voting strategy when both hierarchical and non-hierarchical segmentations are available from the crowd.

The proposed model is generic and we believe that the experimental findings extend beyond query and sentence segmentation to other kinds of linguistic annotations where hierarchical and non-hierarchical schemes co-exist.

Acknowledgment

Thanks to Rishiraj Saha Roy, IIT Kharagpur, for his valuable inputs during this work.

References

Steven P. Abney. 1991. *Parsing By Chunks*. Kluwer Academic Publishers.

Steven P. Abney. 1992. Prosodic Structure, Performance Structure And Phrase Structure. In *Proceedings 5th Darpa Workshop on Speech and Natural Language*, pages 425–428. Morgan Kaufmann.

Steven P. Abney. 1995. Chunks and dependencies: Bringing processing evidence to bear on syntax. *Computational Linguistics and the Foundations of Linguistic Theory*, pages 145–164.

Ron Artstein and Massimo Poesio. 2008. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596.

Kalika Bali, Monojit Choudhury, Diptesh Chatterjee, Sankalan Prasad, and Arpit Maheswari. 2009. Correlates between Performance, Prosodic and Phrase Structures in Bangla and Hindi: Insights from a Psycholinguistic Experiment. In *ICON '09*, pages 101 – 110.

Shane Bergsma and Qin Iris Wang. 2007. Learning Noun Phrase Query Segmentation. In *EMNLP-CoNLL '07*, pages 819–826.

Akshar Bharati, Vineet Chaitanya, and Rajeev Sangal. 1995. *Natural Language Processing: A Paninian Perspective.* Prentice.

Thorsten Brants. 2000. Inter-annotator agreement for a German newspaper corpus. In *In Proceedings of Second International Conference on Language Resources and Evaluation LREC-2000*.

Matthias Hagen, Martin Potthast, Benno Stein, and Christof Bräutigam. 2011. Query segmentation revisited. In *WWW '11*, pages 97–106.

Jian Huang, Jianfeng Gao, Jiangbo Miao, Xiaolong Li, Kuansan Wang, Fritz Behr, and C. Lee Giles. 2010. Exploring Web Scale Language Models for Search Query Processing. In *WWW '10*, pages 451–460.

Dan Jurafsky and James H Martin. 2000. *Speech & Language Processing*. Pearson Education India.

Klaus Krippendorff. 2004. *Content Analysis: An Introduction to Its Methodology*. Sage,Thousand Oaks, CA.

Matthew Lease, Vaughn Hester, Alexander Sorokin, and Emine Yilmaz, editors. 2011. *Proceedings of the ACM SIGIR 2011 Workshop on Crowdsourcing for Information Retrieval (CIR 2011)*.

Rohan Ramanath, Monojit Choudhury, Kalika Bali, and Rishiraj Saha Roy. 2013. Crowd Prefers the Middle Path: A New IAA Metric for Crowdsourcing Reveals Turker Biases in Query Segmentation. In *Proceedings of ACL*. ACL.

Rishiraj Saha Roy, Niloy Ganguly, Monojit Choudhury, and Srivatsan Laxman. 2012. An IR-based Evaluation Framework for Web Search Query Segmentation. In *SIGIR '12*, pages 881–890. ACM.

David Vadas and James R. Curran. 2011. Parsing Noun Phrases in the Penn Treebank. *Comput. Linguist.*, 37(4):753–809, December.