When the full parsing tree of a sentence is available, only the constituents in the tree are considered as argument candidates. In CoNLL-2005, full parsing trees are provided by two full parsers: the Collins parser (Collins, 1999) and the Charniak parser (Charniak, 2000). According to Punyakanok et al. (2005), the boundary agreement of Charniak is higher than that of Collins; therefore, we choose the Charniak parser's results. However, there are two million nodes on the full parsing trees in the training corpus, which makes the training time of machine learning algorithms extremely long. Besides, noisy information from unrelated parts of a sentence could also affect the training of machine learning models. Therefore, our system exploits the heuristic rules introduced by Xue and Palmer (2004) to filter out simple constituents that are unlikely to be arguments. Applying pruning heuristics to the output of Charniak's parser effectively eliminates 61% of the training data and 61.3% of the development data, while still achieves 93% and 85.5% coverage of the correct arguments in the training and development sets, respectively.

2.2 Argument Classification

This stage assigns the final labels to the candidates derived in Section 2.1. A multi-class classifier is trained to classify the types of the arguments supplied by the pruning stage. In addition, to reduce the number of excess candidates mistakenly output by the previous stage, these candidates can be labeled as null (meaning "not an argument"). The features used in this stage are as follows.

Basic Features

- **Predicate** – The predicate lemma.
- **Path** – The syntactic path through the parsing tree from the parse constituent being classified to the predicate.
- **Constituent Type**
- **Position** – Whether the phrase is located before or after the predicate.
- **Voice** – passive: if the predicate has a POS tag VBN, and its chunk is not a VP, or it is preceded by a form of "to be" or "to get" within its chunk; otherwise, it is active.
- **Head Word** – calculated using the head word table described by Collins (1999).
- **Head POS** – The POS of the Head Word.
- **Sub-categorization** – The phrase structure rule that expands the predicate's parent node in the parsing tree.
- **First and Last Word/POS**
- **Named Entities** – LOC, ORG, PER, and MISC.
- **Level** – The level in the parsing tree.

Combination Features

- **Predicate Distance Combination**
- **Predicate Phrase Type Combination**
- **Head Word and Predicate Combination**
- **Voice Position Combination**

Context Features

- **Context Word/POS** – The two words preceding and the two words following the target phrase, as well as their corresponding POSs.
- **Context Chunk Type** – The two chunks preceding and the two chunks following the target phrase.

Full Parsing Features

We believe that information from related constituents in the full parsing tree helps in labeling the target constituent. Denote the target constituent by *t*. The following features are the most common baseline features of *t*'s parent and sibling constituents. For example, Parent/ Left Sibling/ Right Sibling Path denotes *t*'s parents', left sibling's, and right sibling's Path features.

- **Parent / Left Sibling / Right Sibling Path**
- **Parent / Left Sibling / Right Sibling Constituent Type**
- **Parent / Left Sibling / Right Sibling Position**
- **Parent / Left Sibling / Right Sibling Head Word**
- **Parent / Left Sibling / Right Sibling Head POS**
- **Head of PP parent** – If the parent is a PP, then the head of this PP is also used as a feature.

Argument Classification Models