

# Part 5: Structured Prediction Software

# Illinois Structured Learning Library

- ❖ A general purpose learning library in JAVA
- ❖ Learning Algorithms:
  - ❖ Structured Perceptron
  - ❖ Structured SVM
- ❖ Support multi-core learning
- ❖ Can be download at  
<http://cogcomp.cs.illinois.edu/software/illinois-sl>

Other libraries to consider: PyStruct, Vowpal Wabbit, SVMStruct, StructEd, Wolfe, Factorie

# For your own application

You need to implement the following classes/functions

Class/Functions	Explanation	Example
<code>IInstance</code>	Input	Sentence
<code>Istructure</code>	Output	POS tags
<code>getFeatureVector</code>	Feature generator	Emission/Transition Features
<code>InferenceSolver</code>	Inference	Viterbi
<code>getLoss</code>	Loss function	Hamming Loss

# Example: IInstance for POS tagging

```
public class Sentence implements IInstance {  
    public final int[] tokens;  
}
```

# Example: IStructure for POS tagging

```
public class POSTag implements IStructure {  
    public final int[] tags;  
    public POSTag(int[] tags){  
        this.tags = tags;  
    }  
}
```

# Example: Feature Generator

```
public IFeatureVector getFeatureVector(IInstance x, IStructure y) {  
    FeatureVectorBuffer fv = new FeatureVectorBuffer();  
    Sentence ins = (Sentence) x;  
    int[] tags = ((POSTag) y).tags;
```

```
// add emission features
```

```
for (int i = 0; i < ins.tokens.length; i++)  
    fv.addFeature(ins.tokens[i] * lm.getNumOfFeature() + tags[i], 1.0f);
```

Add emission features to  $\mathbf{f}_v$

```
// add prior features
```

```
fv.addFeature(lm.getNumOfFeature() * lm.getNumOfLabels() + tags[0], 1.0f);
```

Add transition features to  $\mathbf{f}_v$

```
// add transition features
```

```
int priorEmissionOffset = lm.getNumOfFeature() * lm.getNumOfLabels() + lm.getNumOfLabels();
```

```
// calculate transition features
```

```
for (int i = 1; i < ins.tokens.length; i++)
```

```
    fv.addFeature(priorEmissionOffset + tags[i - 1] * lm.getNumOfLabels() + tags[i], 1.0f);
```

```
return fv.toFeatureVector();
```

```
}
```

# Example: Feature Generator

```
public IFeatureVector getFeatureVector(IInstance x, IStructure y) {
    FeatureVectorBuffer fv = new FeatureVectorBuffer();
    Sentence ins = (Sentence) x;
    int[] tags = ((POSTag) y).tags;

    // add emission features
    for (int i = 0; i < ins.tokens.length; i++)
        fv.addFeature(ins.tokens[i] + lm.getNumOfFeature() * tags[i], 1.0f);

    // add prior features
    fv.addFeature(lm.getNumOfFeature() * lm.getNumOfLabels() + tags[0], 1.0f);

    // add transition features
    int priorEmissionOffset = lm.getNumOfFeature() * lm.getNumOfLabels() + lm.getNum
    // calculate transition features
    for (int i = 1; i < ins.tokens.length; i++)
        fv.addFeature(priorEmissionOffset + tags[i - 1] * lm.getNumOfLabels() + tags

    return fv.toFeatureVector();
}
```

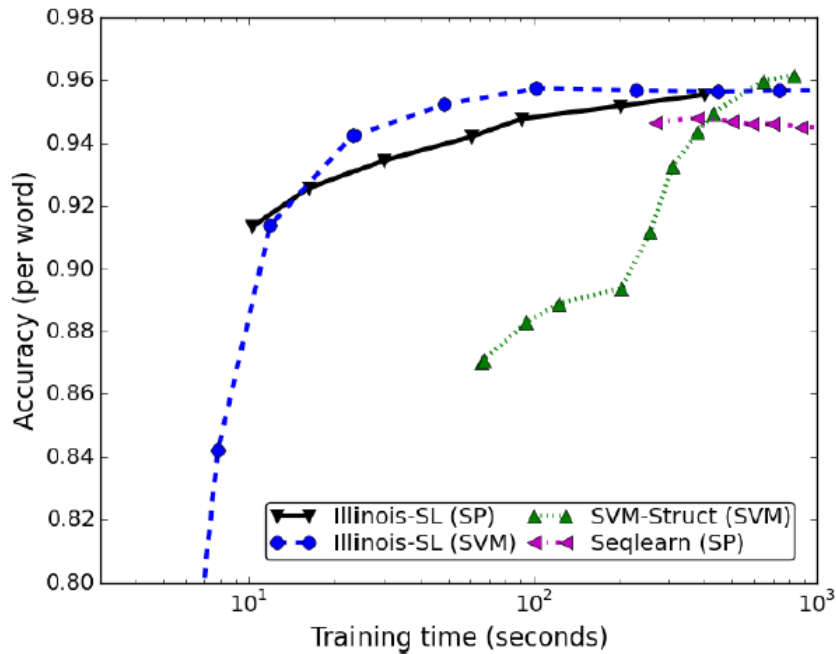
# Built-in applications

Task	Input	Structure	Inference	Features
POS Tagging	sentence	Tag sequence	Viterbi algorithm	Emission and transition features
Dependency Parsing	sentence	Dependency tree	Chu-Liu-Edmonds algorithm	Tree edge features
Cost-sensitive multiclass classification	document	Document category	Simple enumeration	Document features

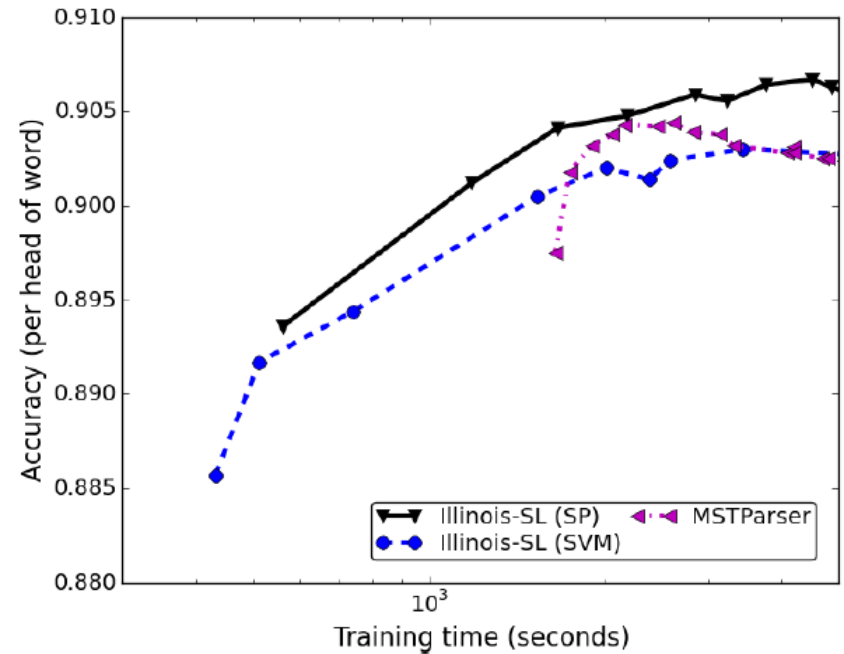


# Performance

## ❖ On standard NLP tasks



(a) Part-of-speech Tagging



(b) Dependency Parsing