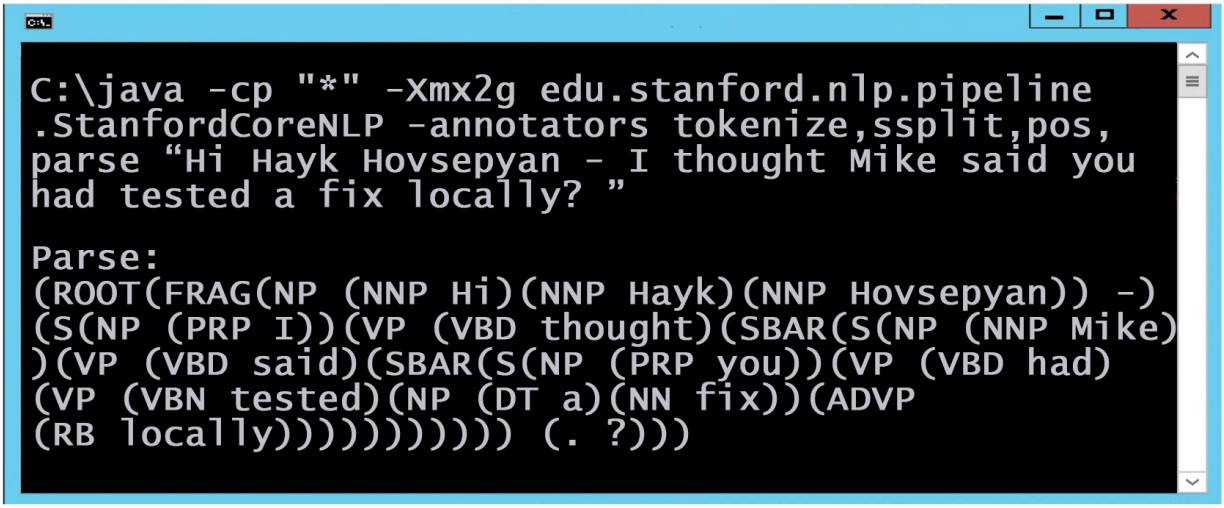
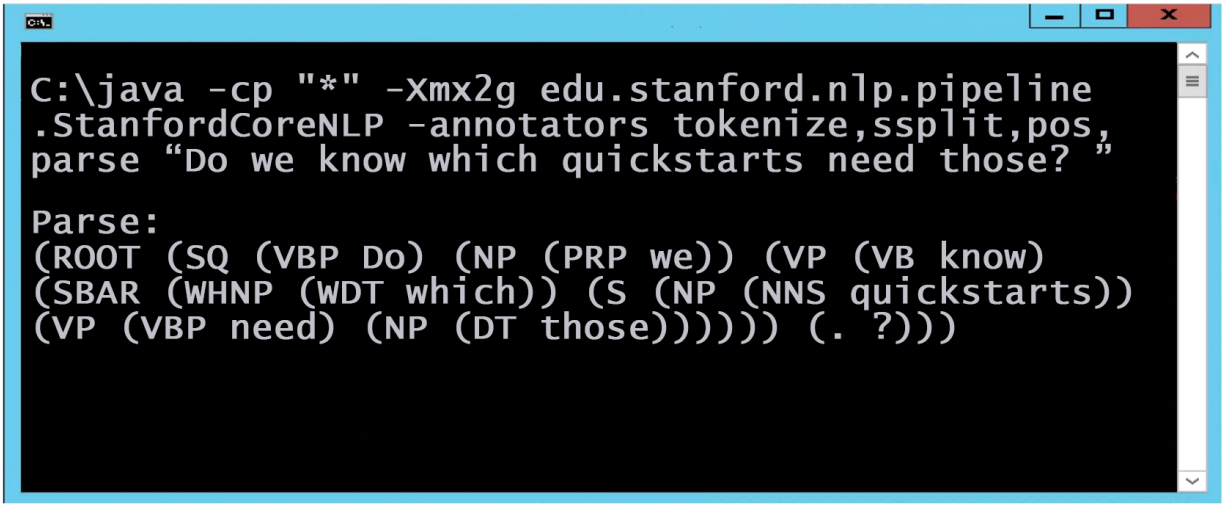
Parameters:

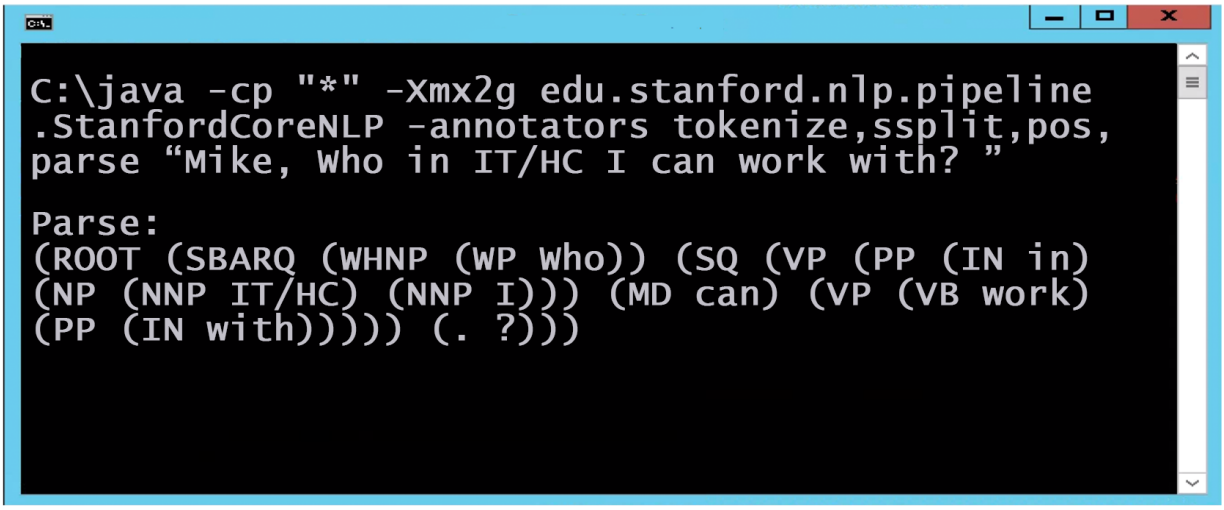
False negative example:



True positive example (SQ):



True positive example (SBARQ):



Ranges of dates for mining the repos

For all 345 open-source projects, we collect and analysis data published between April 06，2001 and August 14, 2017.

For six benchmark datasets the ranges of dates for mining the repos list as following:

Project	From	To
AIRFLOW	April 12, 2016	August 01, 2017
ANY23	October 12, 2011	July 27, 2017
DASHBUILDER	March 25， 2015	July 20, 2017
DROOLS	March 27, 2008	July 17, 2017
IMMUTANT	October 26, 2011	Jun 23, 2017
JBTM	January 03, 2006	Jun 26, 2017

Source code for co-reference:

Following is the java code for co-reference

```
package main;

import java.util.ArrayList;
import java.util.List;
import java.util.Map;
import java.util.Properties;

import edu.stanford.nlp.coref.CorefCoreAnnotations;
import edu.stanford.nlp.coref.data.CorefChain;
import edu.stanford.nlp.coref.data.Mention;
import edu.stanford.nlp.ling.CoreAnnotations;
import edu.stanford.nlp.ling.CoreAnnotations.SentencesAnnotation;
import edu.stanford.nlp.pipeline.Annotation;
import edu.stanford.nlp.pipeline.StanfordCoreNLP;
import edu.stanford.nlp.util.CoreMap;
import java.sql.*;

public class CorefExample extends SQLConnection {
	public static boolean containsCode(String str){
		String[] strs=str.split(" ");
		for(String term:strs){
			if(term.length()>20)
				return true;
		}
		return false;
	}

	public static int getStart(int num){
		if(num>=32200&&num<=32393)
			return 32394;
		if(num>=73584&&num<=74200)
			return 74201;
		if(num>=147473&&num<=150319)
			return 150320;
		if(num>=226955&&num<=228719)
			return 228720;
		return num;
	}

	public static int getEnd(int num){
		if(num>=32200&&num<=32393)
			return 32199;
```

```
		if(num>=73584&&num<=74200)
			return 73583;
		if(num>=147473&&num<=150319)
			return 147472;
		if(num>=226955&&num<=228719)
			return 226954;
		return num;
	}
	public static void main(String[] args) throws Exception {
		/*String strDemo = "The documentation uses the term _binding variable_ 8 times, so bindings seems to be variables too (but of course special ones). I understand that these bindings are not yet cached because of constraint idempotence. However, in my view the implication should be, that especially a non changing result +can+ be cached! Rather than a _good practice_, the documentation says: \r\n" +
					" {quote} \"Property accessors +must+ not change the state of the object in a way that may effect the rules. Remember that the rule engine effectively +caches+ the results of its matching in between invocations to make it faster.\"\r\n" +
					" {quote} \r\n" +
					" So if kind of functional immutability is required and used inside the engine, I would indeed think about caching method calls (with variables) in constraints, as this saves potentially massive processor time and would be much more intuitive.\r\n" +
					" \r\n" +
					" I don't understand your sentence \r\n" +
					" {quote} \"If you can't or desire not to guarantee this property, Drools won't enforce it caching the first value returned.\"\r\n" +
					" {quote} \r\n" +
					" Is it possible, to enforce Drools to cache variable bindings? Drools shouldn't cache values, that change.\r\n" +
					" \r\n" +
					" Let me say that I'm still intrigued about the huge, powerful Drools framework and that we get this as Open Source including this helpful support. Feel free to change the issue-type to a feature request, as the behavior is not strictly incorrect but optimizable.";
		System.out.println( (strDemo));
		System.out.println( "PARSEEEEE");
		System.out.println(WWFunction.fixCommentBody(strDemo));
		System.out.println( "Doneeeeeeeeeeeeeeeeeeeee");*/


		daoIssues daoIss = new daoIssues();
		//(54,67,243,412,145,433)--Done:54,433,243,412,145
		//start from 13190
		int start=173570;
		start = getStart(start);
		int length=1000;
		int end = start+length;
```

```
		end=getEnd(end);
		String sql = "SELECT * from jira_issue where id < "+end+" and id >= "+start+"
and id not in
(173570,120507,120195,106250,105984,86797,84125,84047,76665,73121,73115,70700,62957,
49155,41036,33266,30625,11778,12549, 12836,12847,12869,13190,14474,26014,30544) and
project not in (54,433,243,145,432,67) and issue_type not in (2,29) and id in (select issue from
jira_issue_comment group by issue having count(id) > 1);";

		//String sql = "SELECT * from jira_issue where id = 188442";
				//"SELECT * from jira_issue where id = 73703";//project in
(54,412,145,433) and issue_type not in (2,29) and id not in (32384,228717);";
		// Gets all the issues and their comments
		ArrayList<Issue> features = daoIss.getAllIssuesandComments(sql);
		StandfordNLP nlp = new StandfordNLP();
		System.out.println("Done Loading.");
		while(features.size()>0){
		for (Issue feature : features) {
			//Stanford NLP

			if(!containsCode(feature.getAllCommentsasString())){
			feature = nlp.annotate(feature, false);
			//Save to database
			if (feature != null) {
				daoIss.saveFeatureComments(feature);
			}
			}else{
				System.out.println("Issue:"+feature.getId()+"length:" +
feature.getAllCommentsasString().length());
			}
			//System.out.println("Issue:" + feature.getId());
			/*for (Comment c:feature.getComments()) {
				System.out.println(c.getId());
				System.out.println(c.getBody());
				System.out.println(c.getCoref());
				System.out.println(c.getNumofSentences());
				System.out.println(c.getSentences());
				System.out.println(c.getSentencesPOS());
			}*/
		}
		start=getStart(end);
		end=start+length;
		end = getEnd(end);
		System.out.println("Current start point:"+start);
		sql = "SELECT * from jira_issue where id < "+end+" and id >= "+start+" and id
not in (11778,12549, 12836,12847,12869,13190,14474) and project not in (54,433,243,145,432)
```

```
and issue_type not in (2,29) and id in (select issue from jira_issue_comment group by issue
having count(id) > 1);";
                features = daoIss.getAllIssuesandComments(sql);
                System.out.println("Loading done");
                }
                System.out.println("Doneeeeeeee");


        }
}
```