

```

package com.dlj.ir.analysis.hi;

import java.util.Map;
import java.util.WeakHashMap;

/**
 * Hindi light stemmer- removes number, gender and case suffixes from nouns and adjectives
 * Done by Ljiljana Dolamic (University of Neuchatel, www.unine.ch/info/clef/)
 *
 * @author Ljiljana Dolamic. Email: ljiljana.dolamic@unine.ch
 */
public class HindiStemmerLight{

    /**
     * A cache of words and their stems
     */
    static private Map<String, String> cache = new WeakHashMap<String, String>();

    /**
     * A buffer of the current word being stemmed
     */
    private StringBuilder sb = new StringBuilder();

    /**
     * Default constructor
     */
    public HindiStemmerLight() {
    }

    public String stem(String word) {
        String result = cache.get(word);

        if (result != null)
            return result;

        //
        sb.delete(0, sb.length());

        //
        sb.append(word);

        /* remove the case endings from nouns and adjectives */
        remove_suffix(sb);

        result = sb.toString();
        cache.put(word, result);

        return result;
    }
}

```

```

private void remove_suffix(StringBuilder word) {
    int len = word.length() - 1;

    /* article */

    if (len > 4) {
        if (word.substring( len- 2, len+1).equals("यि") ) {
            word.delete(len-2 , len + 1);
            return;
        }
        if (word.substring( len- 2, len+1).equals("यो") ) {
            word.delete(len-2 , len + 1);
            return;
        }
    }

    /* end if len >4 */
    if (len > 3) {
        if (word.substring(len-1, len+1).equals("ए") ) {
            word.delete(len - 1, len + 1);
            return;
        }
        if (word.substring(len-1, len+1).equals("ओ") ) {
            word.delete(len - 1, len + 1);
            return;
        }
        if (word.substring(len-1, len+1).equals("आ") ) {
            word.delete(len - 1, len + 1);
            return;
        }
        if (word.substring(len-1, len+1).equals("इ") ) {
            word.delete(len - 1, len + 1);
            return;
        }
        if (word.substring( len- 1, len+1).equals("ये") ) {
            word.delete(len-1 , len + 1);
            return;
        }
        if (word.substring(len-1, len+1).equals("े") ) {
            word.delete(len - 1, len + 1);
            return;
        }
        if (word.substring(len-1, len+1).equals("ै") ) {
            word.delete(len - 1, len + 1);
            return;
        }
        if (word.substring( len- 1, len+1).equals("ी") ) {
            word.delete(len-1 , len + 1);
            return;
        }
        if (word.substring(len-1, len+1).equals("ी") ) {
            word.delete(len - 1, len + 1);
        }
    }
}

```

```

        return;
    }
    if (word.substring(len-1, len+1).equals("र")) {
        word.delete(len - 1, len + 1);
        return;
    }
    if (word.substring(len-1, len+1).equals("रै")) {
        word.delete(len - 1, len + 1);
        return;
    }
}
} /* end if len > 3 */
if (len > 2) {
    if (word.substring(len, len+1).equals(" र")) {
        word.delete(len , len + 1);
        return;
    }
    if (word.substring(len, len+1).equals("ँ")) {
        word.delete(len , len + 1);
        return;
    }
    if (word.substring(len, len+1).equals("ं")) {
        word.delete(len , len + 1);
        return;
    }
    if (word.substring(len, len+1).equals("ँ")) {
        word.delete(len , len + 1);
        return;
    }
    if (word.substring(len, len+1).equals("ँ")) {
        word.delete(len , len + 1);
        return;
    }
    if (word.substring(len, len+1).equals("ँ")) {
        word.delete(len , len + 1);
        return;
    }
}
} /* end if len > 2 */
return;
}
}

```