El trabajo práctico es una implementación de un MNLPTK (Minimal Natural Language Processing Tokenizer), que será utilizado para crear un diccionario de datos a partir de un texto de entrada en formato .txt.

El diccionario de datos estará creado utilizando un archivo en formato json. Y además se obtendrá un archivo de output en formato .txt que tendrá detalles sobre cómo se procesó el archivo.

Se tomaron varias decisiones:

1. Cada palabra, por más que sea ambigua, pertenecerá a 1 solo token
2. A la hora de partir el texto para hallar sus componentes, esto se realizará por medio de una expresión regular que también tomará los dígitos apropiadamente
3. Se utilizó una función hash para obtener los valores de los lexemas cuando se analizara uno. En total, para cada uno de los tokens, se creó una tabla hash con los valores necesarios correspondientes
4. Se prescindió de tomar a los patrones como algo separado de los lexemas, porque son lo mismo
5. Se usaron expresiones regulares por extensión, debido a la enorme dificultad de crear expresiones regulares por compresión
6. Se usó un conjunto para mostrar las estadísticas de lexemas nuevos y lexemas hallados porque no era necesario tener ningún tipo de ordenación y solo necesitábamos obtener su tamaño para mostrar las estadísticas, además de que si se repiten lexemas este hecho no incrementa el número general de lexemas en el texto
7. Utilizamos la función hash estándar de Python, que es SipHash, porque según un paper esta tiene un buen rendimiento, suficiente para nuestros propósitos
8. Los errores léxicos se almacenarán como parte del diccionario de datos por si se requiera para uso posterior en otra sección, el usuario los definirá y además que existan estos errores no supondrá un problema para el analizador léxico, pues simplemente tomará el siguiente valor durante el proceso de análisis léxico

Para este trabajo no se aplicaron mejoras de ningún tipo.