Para esta práctica nos hemos servido de las imágenes proporcionadas por el profesor en el enunciado de la práctica y de la libería “math.h” de c++ para calcular.

A partir de dichas imágenes hemos obtenido el siguiente código:

Similitud DFR

```
double Buscador::DFR(int Doc) {
	double sim=0;
	double wiq=0.1,wid=0,ft=0,nt=0,ftd=0,freq=0,ld=0,avr_ld=0;
	InformacionTermino it;
	for(auto
i=getIndiceDocs().begin();i!=getIndiceDocs().end();++i){
		if(i->second.getIdDoc()==Doc){
			ld=i->second.getTamBytes();//tamaño del documento
		}
		avr_ld+=i->second.getTamBytes();
	}
	avr_ld=avr_ld/getIndiceDocs().size();//media de los tamaños de
los documentos de la coleccion

	for(auto
i=getIndicePregunta().begin();i!=getIndicePregunta().end();++i){//Para
cada término de la pregunta
		ft=0;
		nt=0;
		if(getIndice().find(i->first)!=getIndice().end()){
			for(auto j=getIndice().find(i->first)-
>second.getDocs().begin();j!=getIndice().find(i->first)-
>second.getDocs().end();++j){
				if(j->first==Doc){//Si es el documento
					freq=j->second.getFt();//Veces que
aparece el termino en el documento
				}
				ft+=j->second.getFt();//Frecuencia del término
en la coleccion
				++nt;
			}
		}
		else{
			freq=0;
			ft=0;
		}

		ftd=freq*((log10(1+((c*avr_ld)/ld))/log10(2)));

		if(nt!=0){
			wid=((log10(1+(ft/getIndiceDocs().size()))/log10(2))
+
(ftd*(log10((1+(ft/getIndiceDocs().size()))/(ft/getIndiceDocs().size()
))/log10(2)))) * ((ft+1)/(nt*(ftd+1)));
		}
		else{
			wid=0;
		}
		wiq=(double)i->second.getFt()/getIndicePregunta().size();
		sim+=wiq*wid;
	}
	return sim;
}
```

Similitud BM25

```
double Buscador::BM25(int Doc) {
	double sim=0;
	double idf,numpal,medianumpal;
	for(auto
i=getIndiceDocs().begin();i!=getIndiceDocs().end();++i){
		if(i->second.getIdDoc()==Doc){
			numpal=i->second.getNumPalSinParada();
		}
		medianumpal+=i->second.getNumPalSinParada();
	}
	medianumpal=medianumpal/getIndiceDocs().size();

	InformacionTermino it;
	for(auto
i=getIndicePregunta().begin();i!=getIndicePregunta().end();++i){//Para
cada término de la pregunta
		if(getIndice().find(i->first)!=getIndice().end()){
			it=getIndice().find(i->first)->second;
			idf=log10((getIndiceDocs().size()-
(it.getDocs().size()+0.5))/(it.getDocs().size()+0.5));
			if(it.getDocs().find(Doc)!=it.getDocs().end()){
				sim+=fabs(idf*((it.getDocs().find(Doc)-
>second.getFt()*(k1+1))/(it.getDocs().find(Doc)-
>second.getFt()+(k1*(1-b+(b*(numpal/medianumpal)))))));
			}
			else{
				//Numeros absolutos --> fabs(num);
				sim+=fabs(idf*(0*(k1+1))/(0+(k1*(1-
b+(b*(numpal/medianumpal))))));
			}
		}
	}
	return sim;
}
```

Módulo para sacar el nombre de un documento sin extensión ni ruta a partir de su ID

```
string Buscador::SacaNombre(long int id) const {//Devuelve el nombre
sin extension de un documento a partir de una id
	string name;
	list<string> tokens;

	for(auto
it=getIndiceDocs().begin();it!=getIndiceDocs().end();++it){//Obtengo
ruta+nombre+extension
		if(it->second.getIdDoc()==id){
			name=it->first;
		}
	}
	Tokenizador tokenizer("\\/",false,false);
    tokenizer.Tokenizar(name, tokens);
    auto nombre=tokens.end();//Obtiene el último elemento de la ruta
	--nombre;
	name=*nombre;
	tokenizer.DelimitadoresPalabra(".");
	tokenizer.Tokenizar(name,tokens);
	nombre=tokens.begin();
	return *nombre;
}
```

Salida búsqueda

Los archivos muestran una salida demasiado larga como para ser mostrada por lo que se tendrán que calcular cada vez.

Salida trec_eval

BM25

Queryid (Num): All

Total number of documents over all queries

Retrieved: 35109

Relevant: 324

Rel_ret: 321

Interpolated Recall - Precision Averages:

at 0.00 0.1339

at 0.10 0.1325

at 0.20 0.1316

at 0.30 0.1175

at 0.40 0.1048

at 0.50 0.1019

at 0.60 0.0926

at 0.70 0.0841

at 0.80 0.0806

at 0.90 0.0735

at 1.00 0.0733

Average precision (non-interpolated) for all rel docs(averaged over queries)

0.0977

Precision:

At 5 docs: 0.0554

At 10 docs: 0.0458

At 15 docs: 0.0361

At 20 docs: 0.0301

At 30 docs: 0.0241

At 100 docs: 0.0129

At 200 docs: 0.0117

At 500 docs: 0.0077

At 1000 docs: 0.0039

R-Precision (precision after R (= num_rel for a query) docs retrieved):

Exact: 0.0751

DFR

Queryid (Num): All

Total number of documents over all queries

Retrieved: 35109

Relevant: 324

Rel_ret: 321

Interpolated Recall - Precision Averages:

at 0.00 0.1429

at 0.10 0.1356

at 0.20 0.1291

at 0.30 0.1101

at 0.40 0.1079

at 0.50 0.1064

at 0.60 0.0861

at 0.70 0.0746

at 0.80 0.0700

at 0.90 0.0601

at 1.00 0.0600

Average precision (non-interpolated) for all rel docs(averaged over queries)

0.0935

Precision:

At 5 docs: 0.0482

At 10 docs: 0.0398

At 15 docs: 0.0394

At 20 docs: 0.0355

At 30 docs: 0.0293

At 100 docs: 0.0182

At 200 docs: 0.0116

At 500 docs: 0.0077

At 1000 docs: 0.0039

R-Precision (precision after R (= num_rel for a query) docs retrieved):

Exact: 0.0606

Gráfica de comparación de resultados